arXiv:2304.01811v2 [cs.LG] 1 Dec 2023

HarsanyiNet：通过单次前向传播计算精确的Shapley值

Lu Chen [*1] **Siyu Lou** [*1 2] **Keyan Zhang** [1] **Jin Huang** [1] **Quanshi Zhang** [1]

摘要

Shapley值被广泛视为一种可信的归因指标。然而，当人们使用Shapley值来解释深度神经网络（DNN）输入变量的归因时，在实际应用中，通常需要付出极高的计算成本，才能近似得到相对准确的Shapley值。因此，我们提出了一种新颖的网络架构——*HarsanyiNet*，它能够对输入样本进行推理，并在一次前向传播中同时计算输入变量的精确Shapley值。HarsanyiNet的设计基于这样一个理论基础：Shapley值可以重新表述为对网络所编码的Harsanyi交互的重新分配。

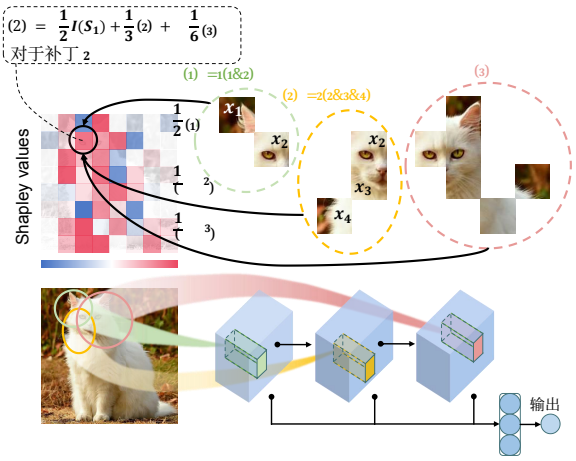


图 *1*。HarsanyiNet 概览。HarsanyiNet 对不同的 Harsanyi交互进行编码，每种交互都表示不同图像块之间的 AND 关系。Shapley值可以通过重新分配 Harsanyi交互来计算。

1. 引言

近年来，可解释人工智能（XAI）受到了广泛关注。解释深度神经网络（DNN）的一种典型方向，是估计某个输入变量（例如图像中的一个像素或句子中的一个词）对网络输出的显著性/重要性/贡献。相关研究被称为归因方法 （Bach 等，2015；Selvaraju 等，2017年；Sundararajan 等，2017年；Lundberg & Lee，2017）。与大多数缺乏坚实理论支持的归因方法相比，Shapley值（Shapley，1953）已被证明是博弈论中唯一满足线性、虚设性、对称性和有效性 公理的解（Young，1985）。因此，Shapley值被广泛认为是衡量每个输入变量归因的一种相对可靠的方法。

然而，在实际应用中使用 Shapley值往往并不现实，因为（1）计算精确的 Shapley值是 NP难；（2）现有的近似技术（Castro 等人，2009年；Lundberg & Lee，2017）常面临两难：要以可接受的准确度近似Shapley值，通常需要进行大量网络推理。

因此，本文旨在直接跳出上述困境，并设计一种名为*HarsanyiNet*的神经网络，使其能够在单次前向传播中同时对输入样本进行模型推理，并计算每个输入变量的精确 Shapley值[1]。

具体而言，HarsanyiNet的理论基础是，输入变量的Shapley值可以重新表述为对深度神经网络所编码的不同Harsanyi交互（Harsanyi，1963）的重新分配。形式上，给定一个深度神经网络和一个包含 n 个变量的输入样本，Harsanyi交互 S 表示 S中的变量之间的一种与关系，该关系由深度神经网络编码。深度神经网络通常会编码许多不同的Harsanyi交互。每种Harsanyi交互都会对深度神经网络的推理得分产生特定的数值

[*]共同贡献 [1]中国上海交通大学 [2]中国 Eastern Institute for Advanced Study。通信作者：Quanshi Zhang 是本文的通信作者，现任职于中国上海交通大学计算机科学与工程系及 John Hopcroft Center。<zqs1022@sjtu.edu.cn>。

论文集收录于 40th 国际机器学习会议，美国夏威夷州檀香山。PMLR 202，2023。版权所有 © 2023，作者所有。

[1]https://github.com/csluchen/harsanyinet

贡献，记为 $I(S)$。以图像块$S=\{$眼睛、鼻子、嘴巴$\}$之间的交互为例。如果所有这些图像块同时出现，它们便会构成一个人脸模式，并对人脸检测的置信度得分产生特定的交互效应 $I(S)$。掩蔽任何一个图像块都会破坏这一模式并消除交互效应，即使 $I(S)=0$。

由于已有证明表明，可以利用Harsanyi交互计算Shapley值，因此，HarsanyiNet中每个中间神经元的激活都被设计为表示特定的Harsanyi交互。这个中间神经元称为*Harsanyi* 单元。这种网络设计使我们能够利用Harsanyi单元的激活得分，得出输入变量的精确Shapley值。

与现有的Shapley值近似方法相比，所提出的HarsanyiNet只需进行一次网络推理，便具有显著优势。

• 现有的近似方法例如DeepSHAP（Lund- berg & Lee，2017年）和FastSHAP（Jethani 等，2021年）会产生误差较大的Shapley值，而HarsanyiNet能够生成**精确的**Shapley值，这一点在理论上有保障，也得到了实验验证。

• 目前，唯一能够通过单次前向传播计算精确Shapley值的已有方法是Shap- Net（Wang 等，2021年）。然而，ShapNet的设计仅能编码至多涉及 k 个输入变量的交互（$k \ll n$，他们设置了 $k=4$），且ShapNet的推理（前向传播）计算成本是传统网络的2^k 倍。作为另一种方案，该研究还提出了另一种网络即深层ShapNet，用于编码更复杂的交互，但它无法保证所计算的Shapley值准确。相比之下，HarsanyiNet不限制交互中输入变量的数量，因此适用范围更广，性能也显著更好。

此外，我们在实验中实现了两种特定的 Harsanyi 网络：由多层感知机（多层感知机）扩展而来的 *Harsanyi-MLP* ，以及基于卷积神经网络（卷积神经网络）开发的 *Harsanyi-CNN* 。

本文的贡献可总结如下。(1) 我们提出了一种新颖的神经网络架构 *HarsanyiNet*，能够在一次前向传播中同时执行模型推理并计算精确的 Shapley值。(2) 遵循 HarsanyiNet 的范式，我们分别针对表格数据和图像数据设计了 Harsanyi-MLP 和 Harsanyi-CNN。(3) HarsanyiNet 不会限制特定交互作用的表示方式，但仍能保证 Shapley值的准确性。

2. 相关工作

估计输入变量的重要性/归因/显著性，是可解释人工智能中的一个典型研究方向。一般而言，以往的归因方法通常基于梯度计算输入变量的归因（Simonyan等，2014；Springenberg 等，2015；Shrikumar 等，2016；Selvaraju 等，2017；Sundararajan 等，2017），通过归因反向传播计算归因（Bach 等，2015；Montavon 等，2017；Shrikumar 等，2017），或基于对输入变量施加的扰动进行计算（Ribeiro 等，2016；Zintgraf 等，2017；Fong & Vedaldi，2017；Lundberg & Lee，2017；Plumb 等，2018；Covert 等，2021；Deng 等，2021；Chen 等人，2022）。

2.1. Shapley值

与其他归因方法不同，Shapley值诞生于博弈论。考虑如下合作博弈：一组 n 参与者 $N=\{1,2,\dots,n\}$相互协作，最终获得奖励 R。Shapley值（Shapley，1953）由此提出，作为一种公平方法，用于将总奖励 R 分配给 n参与者。Shapley值 $\phi(i)$ 定义为从 R 分配给参与者 $i\in N$的组合奖励；我们可以认为，$\phi(i)$ 反映了参与者 i 在博弈中的数值贡献。

定义1（Shapley值）。 输入变量 ϕ（i）的 ***Shapley***值为

$$\phi(i) \coloneqq \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!(n-|S|-1)!}{n!}\left[V(S\cup\{i\})-V(S)\right], \quad (1)$$

其中 V： $2^N \mapsto \mathbb{R}$ 表示奖励函数，即，$\forall S\subseteq N$，V（S）衡量参与者子集 S 参与博弈时获得的奖励。因此，V（$\emptyset$）$=0$，而 V（N）$=R$表示 N中所有参与者赢得的总奖励。

忠实性。 Shapley值被广泛认为是一种相对忠实的归因，因为 Young（1985）已证明，它是满足理想归因的线性、虚设性、对称性 和有效性 公理的唯一博弈论解。详情请参见附录A。

除了估计每个输入变量的重要性外，Shapley值还广泛用于估计整个数据集中的每个数据点的重要性，例如可用于解决数据评估问题（Jia 等人，2019年a；2021年）。

2.2. 计算复杂度与近似精度之间的两难

将 Shapley值应用于现实世界应用时，最大的挑战是 NP难的计算复杂度。根据公式 (1) 和第 3 节，当我们计算

对于 DNN 输入变量的 Shapley值，需要对所有 2^n 个不同的掩码样本进行推理。为了减轻计算负担，已有许多近似方法（Castro 等人，2009年；Lundberg 和 Lee，2017年；Covert & Lee，2021年）被提出。然而，如图 3 所示，要提高 Shapley值的近似精度，通常需要进行更多网络推理（例如，对多达数千个掩码样本进行推理）。

具体而言，一些方法通过采样技术估算 Shapley值（Castro 等人，2009年；Strumbelj & Kononenko，2010年；Okhrati & Lipani，2021 年；Mitchell 等人，2022 年），另一些方法则将 Shapley值的近似转化为加权最小二乘问题（Lundberg & Lee，2017年；Simon & Vincent，2020年；Covert & Lee，2021年）。然而，这些方法都面临一个困境，即要获得更精确的近似结果，就需要付出更高的计算成本。

其他研究通过假设特定的数据分布（Chen 等人，2019 年）、忽略输入变量之间较小的交互作用（Wang 等，2022年），或训练一个解释器模型来直接预测 S hapley值（Jethani 等，2021年），以加快计算速度。然而，这些方法无法生成完全精确的 Shapley值。Wang 等（2021年）提出了 ShapNets。ShapNet 受到限制，只能编码少量变量之间的交互作用（通常少于 4 个）。当 ShapNet 扩展到编码更多变量之间的交互作用时，它便无法再精确估算 Shapley值。相比之下，我们的 HarsanyiNet 能够通过单次前向传播精确计算 Shapley值，且不受编码特定交互作用的限制，因此展现出更强的灵活性和更优的性能。

3. 方法

公式 (1) 中定义的 Shapley值被广泛用于估算深度神经网络中输入变量的归因。我们将深度神经网络视为合作博弈，并将输入变量 $\mathbf{x} = [x_1, x_2, ..., x_n]^\intercal$ 视为参与者，$N = \{1, 2, \ldots, n\}$。$v(\mathbf{x}) \in \mathbb{R}$ 对应网络对$\mathbf{x}$的预测得分[2]。令$\mathbf{x}_S$ 表示一个掩码样本，其中 $N \setminus S$，$S \subseteq N$ 中的输入变量由基线值[3]掩码。这样，我们可以将 S 中的输入变量获得的总收益定义为掩码样本$\mathbf{x}_S$上的推理得分即，$V(S) := v(\mathbf{x}_S) - v(\mathbf{x}_\emptyset)$。因此，公式 (1) 中的 Shapley 值 $\phi(i)$ 衡量了重要性 of 网络预测中的第 i 个输入变量 得分。

[2]与以往研究（Jethani 等，2021年；Wang 等，2022年）一样，如果网络的输出为标量，则 $v(\mathbf{x})$可直接定义为网络输出。如果网络的输出为向量，例如多类别分类，我们可以将 $v(\mathbf{x})$定义为对应于真实类别的输出维度。[3]在以往研究中（Lundberg 和 Lee，2017年；Covert & Lee，2021年），基线值可设为零、不同输入的均值或其他统计值。

3.1. 预备知识：Harsanyi交互

Harsanyi交互（或称 Harsanyi 股息）（Harsanyi，1963年）揭示了为何 Shapley值按公式（1）计算这一问题的本质原因。深度神经网络通常不会在模型推理时独立使用每个输入变量。相反，深度神经网络会对不同输入变量之间的交互作用进行建模，并将这些交互作用视为基本的推理模式。为此，Harsanyi交互 $I(S)$衡量由深度神经网络编码的每个输入变量子集 $S \subseteq N$ 之间的交互效应。

定义2（Harsanyi交互）。**</style)关于**模型输出 $= v_S - v_\emptyset$ L $I\ \emptyset\ = 0.$ （$\mathbf{x}$）的变量集合之间的 *Harsanyi*交互递归定义如下 $I(S) := V(S) - \sum_{L \subsetneq S} I(L)$（$\mathbf{x}$）（$\mathbf{x}$）$- \sum_{L \subsetneq S} I($ $)$ 满足（）：

根据定义2，网络输出可以解释为所有 Harsanyi交互之和，即,$v(\mathbf{x}) - v(\mathbf{x}_\emptyset) = \sum_{S \subseteq N} I(S)$。本质上，每个 Harsanyi交互 $I(S)$ 揭示了集合 S 中所有变量之间的与关系。我们以用于人脸检测的视觉模式$S = \{$眼睛，鼻子、嘴巴$\}$ 为简单示例。如果眼睛、鼻子和嘴巴的图像块同时出现，这三个部分共同出现便构成一种视觉模式，并对人脸分类分数 $v(\mathbf{x})$产生数值贡献 $I(S)$。否则，掩蔽 S 中的任何部分都会使该模式失效并消除交互作用，即，使 $I(S) = 0$。

Grabisch 等（2016年）和 Ren 等（2023a）进一步证明，Harsanyi交互同样满足四个性质，即线性、虚设性、对称性和有效性。

3.2. Harsanyi交互构成Shapley值

我们摆脱了第2节所述的计算复杂度与近似精度之间的困境。定理1使我们能够推导出一种新颖的神经网络架构，即*HarsanyiNet*。该架构利用Harsanyi交互，在一次前向传播中同时执行模型推理并计算精确的 Shapley值。

•HarsanyiNet 的基本要求。 HarsanyiNet的核心思想是让不同的中间层神经元表示不同的Harsanyi交互。稍后我们将证明，可以利用这种网络设计在一次前向传播中计算精确的Shapley值。具体而言，我们将在HarsanyiNet中介绍以下两种设计。

首先，如图 1 所示，HarsanyiNet 在神经网络中包含 L 个级联块。在每个块中，我们添加

一个与运算层，位于线性层和 ReLU 层之间。给定一个输入样本$\mathbf{x}$，令 $z_u^{(l)}(\mathbf{x})$表示第 u个线性层中输出特征向量$\mathbf{z}^{(l)}(\mathbf{x})$的第 $\in \mathbb{R}^{m^{(l)}}$ 维。HarsanyiNet 的设计目标是使每个特征维度 $z_u^{(l)}(\mathbf{x})$满足以下两个要求，而 $z_u^{(l)}(\mathbf{x})$也称为*Harsanyi* 单元。定理 3 将说明如何利用 Harsanyi 单元直接计算精确的 Shapley值。

要求1 (R1)。 神经元输出 $z_u^{(l)}(\mathbf{x})$完全由一组特定的输入变量$\mathcal{R}_u^{(l)} \subseteq N$决定，即神经元 $z_u^{(l)}(\mathbf{x})$的感受野。换言之， $N \setminus \mathcal{R}_u^{(l)}$中的其他输入变量都不会影响神经元激活，即，对于任意两个样本$\mathbf{x}$ 和 $\mathbf{x}'$，如果 $\forall\, i \in \mathcal{R}_u^{(l)}$, $x'_i = x_i$，那么 $z_u^{(l)}(\mathbf{x}') = z_u^{(l)}(\mathbf{x})$。

要求2 (R2)。 对神经元 $z_u^{(l)}$感受野 $\mathcal{R}_u^{(l)}$中的任意变量进行掩蔽，都会使 $z_u^{(l)}(\mathbf{x}) = 0$。具体而言，令$\mathbf{x}_S$ 表示在样本$\mathbf{x}$中对集合 $N \setminus S$ 内的变量进行掩蔽后得到的样本。那么，对于任意掩码样本$\mathbf{x}_S$，神经元 $z_u^{(l)}$必须满足性质$z_u^{(l)}(\mathbf{x}_S) = z_u^{(l)}(\mathbf{x}) \cdot \prod_{i\in\mathcal{R}_u^{(l)}} \mathbb{1}(i \in S)$。

这两项要求表明，Harsanyi 单元$z_u^{(l)}$必须表示输入变量之间的与关系，位于 $\mathcal{R}_u^{(l)}$。改变感受野 $\mathcal{R}_u^{(l)}$之外的变量不会影响 Harsanyi 单元的神经元激活，即， $z_u^{(l)}(\mathbf{x}_S) = z_u^{(l)}(\mathbf{x})$；但掩蔽$\mathcal{R}_u^{(l)}$中的任何变量都会使该单元失活，即，令 $z_u^{(l)}(\mathbf{x}_S) = 0$。

其次，尽管 HarsanyiNet 可以具有多种类型的输出（包括标量输出、向量化输出、矩阵输出和张量输出），**网络输出的每个维度都被设计为所有 Harsanyi单元的加权和**。令 $\mathbb{v}(\mathbf{x})$ 表示多维输出，并令 $v(\mathbf{x})$ 表示一个由 $\{\mathbf{w}_v^{(l)}\}$ 参数化的任意输出维度。于是，我们得到

$$\mathbb{v}(\mathbf{x}) = [v(\mathbf{x}), v'(\mathbf{x}), \dots]^\intercal, \quad v(\mathbf{x}) = \sum_{l=1}^{L} (\mathbf{w}_v^{(l)})^\intercal \mathbf{z}^{(l)}(\mathbf{x}), \tag{2}$$

其中，$\mathbf{w}_v^{(l)} \in \mathbb{R}^{m^{(l)}}$ 表示特定输出维度 v 的权重。如图 1 所示，上述公式可以通过添加跳跃连接，将所有 L 层中的 Harsanyi 单元连接到 HarsanyiNet 的输出端来实现。

• **证明我们可以通过单次前向传播计算精确的 Shapley值**。前面的段落仅概述了 Harsanyi 单元的两个要求，第3.3节将介绍如何促使神经元满足这些要求。在此之前，我们推导定理 3，以证明上述要求使我们能够通过一次前向传播计算精确的 Shapley值。

定理1 （Shapley值与Harsanyi交互之间的联系，证明见（Harsanyi，1963））。 该*Shapley*值 $\phi(i)$ 等于包含 i的*Harsanyi*交互均分后的总和，即：

$$\phi(i) = \sum\nolimits_{S\subseteq N: S\ni i} \frac{1}{|S|} I(S). \tag{3}$$

定理1表明，我们可以将 Shap- ley value $\phi(i)$理解为对每个包含变量 i的 Harsanyi交互 $I(S)$进行均匀分配。例如，考虑一个玩具模型，该模型使用年龄 *(a)*、教育程度 *(e)*、职业 *(o)*和婚姻状况 *(m)* 来推断收入水平。为简化说明，我们假设只能分解出四个非零的 Harsanyi交互，即即，$v(\mathbf{x}) = \sum_{S\subseteq N=\{a,e,o,m\}} I(S) = I(\{a,o\}) + I(\{a,e\}) + I(\{a,o,m\}) + I(\{o,m\})$。我们将数值贡献 $I(\{a,o,m\})$均匀分配给年龄、职业和婚姻状况，每个变量都获得 $\frac{1}{3}I(\{a,o,r\})$，作为其归因的一部分。这样，每个输入变量都会累积来自不同 Harsanyi交互的组合归因，例如例如，$\phi(a) = \frac{1}{2}I(\{a,o\}) + \frac{1}{2}I(\{a,e\}) + \frac{1}{3}I(\{a,o,m\})$。累积的归因 $\phi(a)$等于 Shapley值 $\phi(a)$。

引理1 (Harsanyi 单元的Harsanyi交互，证明见附录 C)。 令 *Harsanyi* 单元 $z_u^{(l)}(\mathbf{x})$ 的输出为奖励。然后，令 $J_u^{(l)}(S)$ 表示关于函数 $z_u^{(l)}(\mathbf{x})$的 *Harsanyi*交互。根据要求 *R1* 和 *R2*，有 $J_u^{(l)}(\mathcal{R}_u^{(l)}) = z_u^{(l)}(\mathbf{x})$，且 $\forall S \neq \mathcal{R}_u^{(l)}, J_u^{(l)}(S) = 0$。

定理2 (证明见附录 B)。 设网络输出$v(\mathbf{x}) \in \mathbb{R}$ 表示为 $v(\mathbf{x}) = \sum_{l=1}^{L} (\mathbf{w}_v^{(l)})^\intercal \mathbf{z}^{(l)}(\mathbf{x})$，根据公式 (2)。这样，在网络输出 $v(\mathbf{x})$ 上计算得到的集合 S 中输入变量之间的 *Harsanyi* 交互可以表示为 $I(S) = \sum_{l=1}^{L}\sum_{u=1}^{m^{(l)}} w_{v,u}^{(l)} J_u^{(l)}(S)$。

定理2表明，关于网络输出 $v(\mathbf{x})$ 的 Harsanyi交互$I(S)$关于可以表示为在不同 Harsanyi 单元 $z_u^{(l)}(\mathbf{x})$ 上计算得到的 Harsanyi交互 $J_u^{(l)}(S)$ 之和。通过这种方式，我们将引理1和定理2中的结论代入公式 (3)，得出以下定理。

定理 3 (从中间层的 Harsanyi 单元推导 Shapley值，证明见附录 B)。*Shapley*值 $\phi(i)$可计算如下

$$\phi(i) = \sum\nolimits_{l=1}^{L} \sum\nolimits_{u=1}^{m^{(l)}} \frac{1}{|\mathcal{R}_u^{(l)}|} w_{v,u}^{(l)} z_u^{(l)}(\mathbf{x}) \mathbb{1}(\mathcal{R}_u^{(l)} \ni i). \tag{4}$$

定理 3 证明，Shapley值 $\phi(i)$ 可以在前向传播中，直接利用中间层的 Harsanyi单元$\mathbf{z}^{(l)}(\mathbf{x})$ 的输出计算。

HarsanyiNet 的计算成本。 我们进行一次网络推理，以获取所有 Harsanyi 单元 $z_u^{(l)}(\mathbf{x})$ 的输出。然后，我们根据公式 (4) 计算 Shapley值，其计算成本为 $\mathcal{O}(nM)$，其中 $M=\sum_{l=1}^{L} m^{(l)}$表示 Harsanyi 单元的总数。与一次前向传播的高昂计算成本相比，计算成本 $\mathcal{O}(nM)$ 可以忽略不计。 **因此，我们可以大致将计算Shapley值的总成本视为一次前向传播**。

3.3. 面向 R1 和 R2 设计 HarsanyiNet

本小节首先介绍 HarsanyiNet 的详细设计。基本思路是构建一种神经网络，使每个神经元表示前一层中其子节点之间的与关系，并且该神经元的感受野可以通过取其子节点感受野的并集来计算。接着，定理4证明，这种网络设计满足第 3.2 节中的要求 R1 和 R2。

Harsanyi 块。 如图 1 所示，HarsanyiNet 包含 L 个级联的*Harsanyi* 块。 具体而言，我们使用元组 (l,u) 表示第 u 个 Harsanyi 块的线性层中的第 l 个神经元。每个神经元 (l,u) 都有一组子节点 $\mathcal{S}_u^{(l)}$。 $\mathcal{S}_u^{(l)}$中的子节点可以从前面所有 $(l-1)$ 个 Harsanyi 块的所有神经元中选取[4]。或者，作为一种简化实现，我们也可以只从第 $(l-1)$ 个块中选取子节点。每个神经元 (l,u) 的可学习子节点 $\mathcal{S}_u^{(l)}$ 将在后文介绍。因此，给定子节点集合 $\mathcal{S}_u^{(l)}$ 后，神经元 (l,u) 的神经激活 $z_u^{(l)}(\mathbf{x})$ 通过线性、与和 ReLU 操作计算得到

$$g_u^{(l)}(\mathbf{x}) = (\mathbf{A}_u^{(l)})^\intercal \cdot \left(\mathbf{\Sigma}_u^{(l)} \cdot \mathbb{z}^{(l-1)}\right). \quad \begin{matrix}//\text{Linear operation}\\ \text{on children nodes}\end{matrix} \tag{5}$$

$$h_u^{(l)}(\mathbf{x}) = g_u^{(l)}(\mathbf{x}) \cdot \prod_{(l',u')\in\mathcal{S}_u^{(l)}} \mathbb{1}(z_{u'}^{(l')}(\mathbf{x}) \neq 0). \quad \begin{matrix}//\text{AND}\\ \text{operation}\end{matrix} \tag{6}$$

$$z_u^{(l)}(\mathbf{x}) = \text{ReLU}(h_u^{(l)}(\mathbf{x})). \quad //\text{Non-linear operation} \tag{7}$$

在上述公式中，$\mathbb{z}^{(l-1)} = [\mathbf{z}^{(1)}(\mathbf{x})^\intercal,\ \mathbf{z}^{(2)}(\mathbf{x})^\intercal,\ .....,\ \mathbf{z}^{(l-1)}(\mathbf{x})^\intercal]^\intercal \in \mathbb{R}^{M^{(l)}}$ 将前面所有 $(l-1)$ 个块中的神经元向量化。子节点集合 $\mathcal{S}_u^{(l)}$通过二值对角矩阵 $\mathbf{\Sigma}_u^{(l)} \in \{0,1\}^{M^{(l)}\times M^{(l)}}$实现，该矩阵从前面所有 $(l-1)$ 个块的全部神经元中选择神经元 $(l,\ u)$ 的子节点 $M^{(l)} = \sum_{l'=1}^{l-1} m^{(l')}$ 。 $\mathbf{A}_u^{(l)} \in \mathbb{R}^{M^{(l)}}$ 表示权重向量。

公式 (5) – (7) 中的三个操作确保每个 Harsanyi 单元 $z_u^{(l)}(\mathbf{x})$表示一种与关系

[4]具体而言， $\mathcal{S}_u^{(1)}$中的子节点直接从输入变量中选取。

在其子节点中（更多讨论见附录 D）。

为了实现公式 (5) 中的子节点选择，我们通过设置 $(\mathbf{\Sigma}_u^{(l)})_{i,i}=\mathbb{1}((\boldsymbol{\tau}_u^{(l)})_i > 0)$ 来计算二值对角矩阵 $\mathbf{\Sigma}_u^{(l)}$，其中 $\boldsymbol{\tau}_u^{(l)} \in \mathbb{R}^{M^{(l)}}$ 是一个可训练参数向量。请注意，在训练阶段，损失函数的梯度无法通过上述实现中的$\mathbf{\Sigma}_u^{(l)}$传递到 $\boldsymbol{\tau}_u^{(l)}$；因此，我们采用直通估计器（STE）（Bengio 等人，2013）来训练参数 $\boldsymbol{\tau}_u^{(l)}$。在前向传播过程中，STE 使用 $(\mathbf{\Sigma}_u^{(l)})_{i,i} = \mathbb{1}((\boldsymbol{\tau}_u^{(l)})_i > 0)$，并在反向传播过程中设置 $\partial(\mathbf{\Sigma}_u^{(l)})_{i,i}/\partial(\boldsymbol{\tau}_u^{(l)})_i = \beta e^{-(\boldsymbol{\tau}_u^{(l)})_i}/(1+e^{-(\boldsymbol{\tau}_u^{(l)})_i})^2$，其中 β 是一个正标量。此外，为了降低公式（6）中与运算的优化难度，我们将与运算近似为 $h_u^{(l)}(\mathbf{x}) = g_u^{(l)}(\mathbf{x})\left[\prod_{u'=1}^{M^{(l)}} (\mathbf{\Sigma}_u^{(l)}\cdot\tanh(\gamma\cdot\mathbf{\Sigma}_u^{(l)}\mathbf{z}^{(l-1)}) + (\mathbf{I}-\mathbf{\Sigma}_u^{(l)})\cdot\mathbf{1})_{u'}\right]^{1/\mathbf{tr}(\mathbf{\Sigma}_u^{(l)})}$，其中 γ 是一个正标量。这里，函数 tanh(·) 的每个输出维度都在 $[0,1)$ 范围内，因为 $\mathbf{z}^{(l-1)}$ 经过了 ReLU 操作，并且 $\forall u, \mathbf{z}^{(l-1)}_u \geq 0$。

输入与感受野。 令$\mathbf{z}^{(0)} = \mathbf{x} - \mathbf{b} \in \mathbb{R}^n$ 为第一个 Harsanyi 块中线性运算的输入，并将基线值 b_i 定义为每个输入变量 x_i的掩码状态。为进一步简化实现，我们采用单一值基线$\mathbf{b}$[5]。值得注意的是，在（Lundberg & Lee，2017；Covert 等人，2020；Sundararajan 和 Najmi，2020；Chen 等人，2022）中已经讨论了更复杂的基线值。根据公式（5）–（7），我们可以得到 Harsanyi 单元（$l,\ u$）的感受野$\mathcal{R}_u^{(l)}$可以递归计算如下。

$$\mathcal{R}_u^{(l)} := \cup_{(l',u')\in\mathcal{S}_u^{(l)}} \mathcal{R}_{u'}^{(l')}, \quad s.t.\ \mathcal{R}_u^{(1)} := \mathcal{S}_u^{(1)}. \tag{8}$$

定理 4 (证明见附录 B)。 根据公式 (5)–(7)，神经元 $z_u^{(l)}$ 的感受野 $\mathcal{R}_u^{(l)}$ 自动满足 *R1* 和 *R2* 两项要求。

该定理证明，根据公式（5）–（7）设置每个神经元 $z_u^{(l)}$，能够成功编码输入变量 $\mathcal{R}_u^{(l)}$之间的 AND 关系。换言之，只有感受野 $\mathcal{R}_u^{(l)}$中的输入变量才能影响神经元输出 $z_u^{(l)}(\mathbf{x})$，而对$\mathcal{R}_u^{(l)}$中的任何输入变量进行掩蔽，都将使 $z_u^{(l)}(\mathbf{x}) = 0$。具体而言，我们以对掩码样本$\mathbf{x}_S$进行推理为例。根据定理4，掩码样本$\mathbf{x}_S$ 通过设置 $\forall i \notin S, x_i = b_i$来实现。随后，该掩码样本只能激活满足$\mathcal{R}_u^{(l)} \subseteq S$条件的所有 Harsanyi 单元。所有其他 Harsanyi 单元均未被激活，即，$\forall \mathcal{R}_u^{(l)} \not\subseteq S, z_u^{(l)}(\mathbf{x}_S) = 0$。

[5]由于我们使用 ReLU 函数作为非线性运算，因此直接将基线值设为 $\mathbf{b} = \mathbf{0}$。

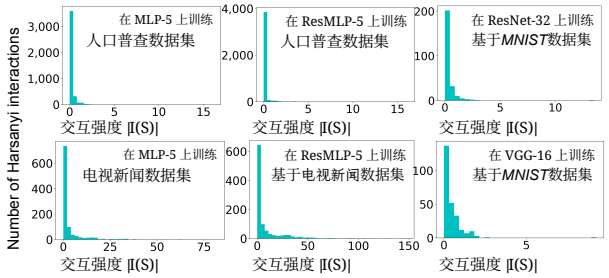


图2。深度神经网络编码的不同 Harsanyi交互的交互强度直方图 $|I(S)|$ 。

3.4. 关于 Harsanyi交互 稀疏性的讨论

理论上，一个模型最多可以编码 2^n 种不同的 Harsanyi交互，但每个 AI 模型在编码交互作用方面都有自身的局限，其实际能够编码的数量远低于 2^n。HarsanyiNet 最多能够学习 $M=\sum_{l=1}^{L} m^{(l)}$ 种 Harsanyi交互，这一点已由引理1和定理2证明。因此，接下来的问题是：HarsanyiNet 是否具有足够的表征能力来处理现实世界中的应用?

为此，近期研究观察到（Deng等，2022；Ren等，2023a；Li和Zhang，2023），并从数学上证明了（Ren 等，2023b），传统深度神经网络在实际应用中往往只编码少数 Harsanyi交互，而非学习所有 2^n Harsanyi交互。具体而言，网络输出可表示为

$$v(\mathbf{x}) = \sum\nolimits_{S\in 2^N} I(S) = \sum\nolimits_{S\in\Omega} I(S) + \epsilon, \qquad (9)$$

其中，$\Omega \subseteq 2^N = \{S' \subseteq N\}$ 表示一个较小的 Harsanyi交互集合，其交互强度 $|I(S)|$相当显著。其他所有 Harsanyi交互的交互强度都可忽略不计，即，$|I(S)| \approx 0$，因此可视为噪声推理模式。$\epsilon = \sum_{S'\in 2^N\setminus\Omega} I(S')$ 相对较小。

此外，我们还进行了新的实验，以验证深度神经网络中 Harsanyi交互的稀疏性。给定一个训练好的网络 v 和一个输入样本 $\mathbf{x}$，我们计算了所有 2^n Harsanyi交互相对于所有$S \subseteq N$的交互强度 $|I(S)|$ 。我们按照（Ren等，2023a）的方法计算不同 Harsanyi交互[6]的交互强度 $|I(S)|$ 。更多细节请参见附录F.6。图2展示了提取出的 Harsanyi交互。我们在各种深度神经网络上进行了此类实验，包括多层感知机、Touvron等人（2022）使用的残差多层感知机[7]、具有32层的残差网络（ResNet-32）（He 等，2016）和具有16层的VGG网络（VGG-16）（Simonyan等，2014）；实验所用数据集包括人口普查收入（Dua & Graff，2017）、电视新闻商业广告检测（Dua & Graff，2017）以及 MNIST（LeCun & Cortes，2010）数据集。检测（Dua & Graff，2017）和 MNIST（LeCun & Cortes，2010）数据集。结果发现，只有少数 Harsanyi交互较为显著。大多数 Harsanyi交互都接近于零，可以视为噪声。

因此，上述实验表明，许多应用只需要深度神经网络编码少数显著的 Harsanyi交互，而不必对指数数量的 Harsanyi交互进行建模。从这个角度来看，HarsanyiNet 具有足够的表征能力。

4. 实验

4.1. 两种类型的 Harsanyi 网络

在实验中，我们按照公式 (5)–(7) 中的范式构建并测试了两种类型的 Harsanyi 网络，即，采用全连接层构建的 HarsanyiNet，即 *Harsanyi-MLP*，以及采用卷积层构建的 HarsanyiNet，即 *Harsanyi-CNN*。Harsanyi-MLP 适用于处理表格数据，而 Harsanyi-CNN 则专为图像数据设计。

• **Harsanyi-MLP**被设计为多层感知机网络的扩展。如第3.3节开头所述，我们没有将每个神经元 (l,u) 与所有 $(l-1)$ 个先前块中的神经元相连接。相反，我们仅从第 $(l-1)$ 个块中选取一组子节点 $\mathcal{S}(l)_u$ 。具体而言，我们通过将 $\boldsymbol{\tau}(l)_u$ 中对应于第 1、2、…、$(l-2)$ 个块中所有神经元的元素固定为 0 来实现这一点。

• Harsanyi-CNN主要采用以下两种具体设置，将卷积层适配到HarsanyiNet范式中。设置1：与 Harsanyi-MLP类似，Harsanyi-CNN根据第$l-1$个块中的神经元，为每个神经元 (l,u) 构建子节点集合 $\mathcal{S}(l)_u$ 。令$C\times K\times K$表示卷积核的张量大小。如图6所示，我们从 $C\times K\times K$子张量中的神经元中，选取神经元 $(l,u=(c,h,w))$的子节点 $\mathcal{S}(l)_u$ 。该子张量从第 $l-1$个块的特征张量中裁剪得到，对应于上层神经元 (l,u) ，其中 c,h,w 表示神经元 (l,u) 的位置。因此，我们得到的 $\boldsymbol{\tau}(l)_u$ 是一个 CK^2维向量。

设定2。此外，我们将位于同一位置、但处于不同通道的所有神经元 $(l,u=(:,h,w))$ 设为共享同一个子节点集合 $\mathcal{S}^{(l)}_{u=}(:,h,w)$，以减少参数数量 $\boldsymbol{\tau}^{(l)}_u$。这可以通过让所有神经元 $(l,u=(1,h,w)),\ldots,(l,u=(C,h,w))$共享同一个参数 $\boldsymbol{\tau}^{(l)}_u$来实现。基于上述设计，我们证明了位于同一位置 (h,w) 、处于不同通道 $(c=1,\ldots,C)$ 的所有 Harsanyi 单元 $(l,u=(c,h,w))$ 具有相同的感受野 $\mathcal{R}^{(l)}_{u=}(:,h,w)$，并且对应于同一个 Harsanyi交互 $I(S=\mathcal{R}^{(l)}_{u=}(:,h,w))$。

[6]对于图像数据，Ren 等（2023a）通过随机采样图像区域来计算 Harsanyi 交互，以降低计算成本。[7]我们分别使用了 5 层多层感知机（MLP-5）和 5 层残差多层感知机（ResMLP-5），每个隐藏层均包含 100 个神经元。

证明请参见附录 E。因此，我们进一步将位于同一位置（h, w）、处于不同通道的神经元视为一个 Harsanyi 单元。这样，公式 (6) 可以改写为 $h_u^{(l)}(\mathbf{x}) = g_u^{(l)}(\mathbf{x}) \cdot \prod_{(l-1,u')\in\mathcal{S}_u^{(l)}} \mathbb{1}(\sum_{c=1}^{C} |z_{u'=(c,h,w)}^{(l-1)}(\mathbf{x})| \neq 0)$.

在实现中，我们首先对输入图像依次应用卷积层、最大池化层和 ReLU 层，以获得中间层中的特征$\mathbf{z}^{(0)}$。随后，我们将$\mathbf{z}^{(0)}$ 视为 Harsanyi-CNN 的输入，而不是直接使用原始像素值作为输入变量。这是因为使用 ReLU 操作使我们能够简单地将 $b_i = 0$ 定义为所有特征维度$\mathbf{z}^{(0)}$的基线值（即，掩码状态）。此外，根据设定2，我们可以将位置（h, w）处的特征向量$\mathbf{z}^{(0)}{}_{(:\,h,\,w)}$视为单个输入变量，并使用$\mathbf{z}^{(0)}{}_{(:\,h,\,w)} = \mathbf{0}$来标识其掩码状态。

4.2. 实验与比较

数据集。我们在 UCI 机器学习知识库中的三个表格数据集（Dua和Graff，2017）上训练了 Harsanyi-MLP，包括人口普查收入数据集（$n = 12$）、酵母数据集（$n = 8$）和电视新闻广告检测数据集（$n = 10$），其中 n 表示输入变量的数量。为简便起见，这些数据集分别称为人口普查、酵母和电视新闻。我们在两个图像数据集上训练了 Harsanyi-CNN：MNIST 数据集（LeCun & Cortes，2010年）和 CIFAR-10 数据集（Krizhevsky 等人，2009年）。

Shapley值的准确性和计算成本。 我们进行了实验，以验证 HarsanyiNet 能否通过一次前向传播计算出准确的 Shapley值。我们评估了 Shapley值计算的准确性和耗时。我们计算了估计的 Shapley值 $\boldsymbol{\phi}_{\mathbf{x}}$ 与真实 Shapley值之间的均方根误差（RMSE），即，RMSE=$\mathbb{E}_{\mathbf{x}}[\frac{1}{\sqrt{n}}||\boldsymbol{\phi}_{\mathbf{x}} - \boldsymbol{\phi}_{\mathbf{x}}^*||]$；对于样本 $\mathbf{x}$，当 $n \leq 16$时，可以根据定义1直接计算真实 Shapley值向量 $\boldsymbol{\phi}_{\mathbf{x}}^*$。

与近似方法进行比较。我们比较了 HarsanyiNet 计算出的 Shapley 值与多种近似方法估算出的结果的准确性，包括采样方法（Castro 等人，2009年）、KernelSHAP（Lundberg & Lee，2017年）、采用配对采样的 KernelSHAP（KernelSHAP-PS）（Covert & Lee，2021年）、对偶采样（Mitchell 等人，2022年）、DeepSHAP（Lundberg & Lee，2017年）和 FastSHAP（Jethani 等，2021年）。为公平比较，这些近似方法也在 HarsanyiNet 上计算 Shapley 值。图 3 显示，许多近似方法通过进行更多次推理，能够生成更准确的 Shapley 值。推理次数被广泛用于衡量近似 Shapley 值的计算成本（Lundberg & Lee，2017年；Ancona 等，2019年），这些方法通常需要进行数千次网络推理才能计算出相对准确的 Shapley 值。相比之下，HarsanyiNet 只需一次前向传播即可获得精确的 Shapley 值（见图 3 中的“⋆”）。DeepSHAP 和 FastSHAP 也能通过一次前向传播计算 Shapley 值，但如表 2 所示，它们估算的 Shapley 值误差远大于 HarsanyiNet。

表 *1.*估计 Shapley值的均方根误差以及深度神经网络的分类准确率。

	HarsanyiNet	浅层 ShapNet	Deep ShapNet[8]
MNIST 数据集			
分类准确率（↑）	**99.16**	40.18	93.85
Shapley值误差（↓）	**1.19e-07**	3.79e-07	0.891
CIFAR-10 数据集			
分类准确率（↑）	**89.34**	20.48	73.51
Shapley值误差（↓）	**6.88e-08**	2.41e-07	0.409
人口普查数据集			
分类准确率（↑）	84.57	84.14	**84.72**
Shapley值误差（↓）	**2.18e-08**	5.12e-07	0.412
酵母数据集			
分类准确率（↑）	**59.91**	57.17	59.70
Shapley值误差（↓）	**3.36e-08**	1.97e-07	0.127
电视新闻数据集			
分类准确率（↑）	82.20	79.72	**82.46**
Shapley值的误差（↓）	**5.69e-08**	2.47e-07	0.239

表 *2.* 人口普查、酵母和电视新闻数据集上计算所得 Shapley值的误差。

数据集 \ 模型	HarsanyiNet	DeepSHAP	FastSHAP
人口普查	**2.18e-08**	0.701	0.270
酵母	**3.36e-08**	1.311	0.467
电视新闻	**5.69e-08**	0.758	0.526

与 ShapNets 进行比较。此外，我们还比较了两种 ShapNet（Wang 等，2021年）的分类准确率和 Shapley值准确率，分别是浅层*ShapNet*和深层*ShapNet*。MNIST 和 CIFAR-10 数据集中的输入样本包含更多输入变量。为了通过定义1计算真实夏普利值，我们随机选取 $n = 12$ 个变量作为样本$\mathbf{x}$前景中的输入变量。这样，通过掩蔽选定的 12 个变量，并将所有其他变量保留为其原始值，计算出真实夏普利值。类似地-

[8]结果使用原论文发布的代码获得（Wang 等，2021年）。具体而言，对于图像数据集，每项实验均采用不同的随机初始化运行十轮，并报告十次运行中的最佳结果。

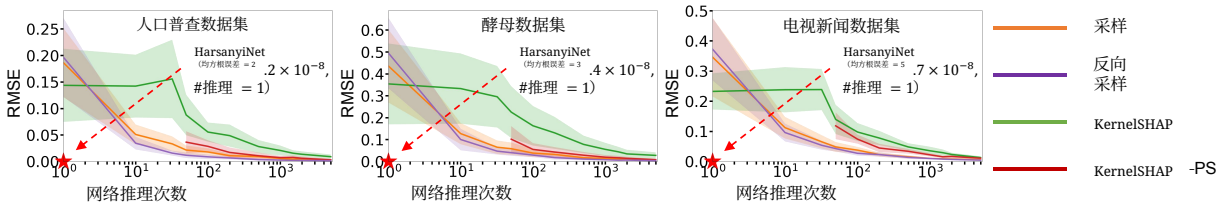


图 *3.*不同方法所需的估计误差和计算成本（网络推理次数）比较。

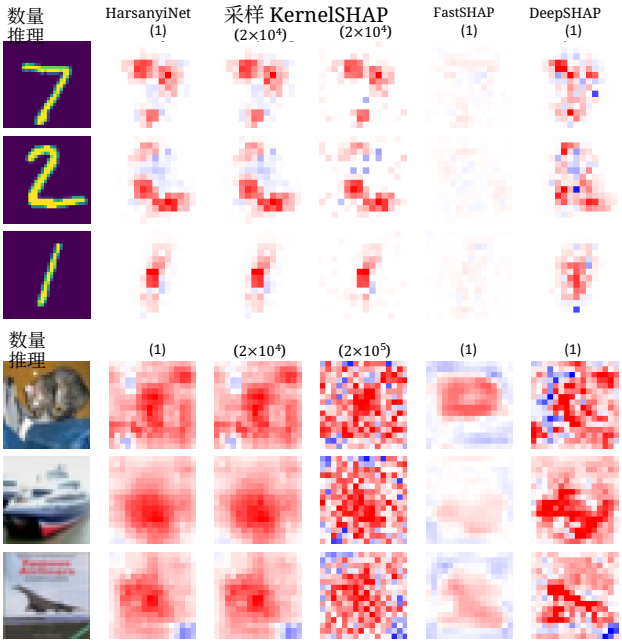


图*4.*不同方法计算得到的 Shapley值[2]。同时展示了近似计算所进行的推理次数。

同样地，当我们仅考虑 $n = 12$输入变量时，仍然可以使用Harsanyi-CNN和ShapNets推导Shapley值（详情请参见附录F.8）。

表 1 显示，HarsanyiNet 和浅层ShapNet 都生成了精确的 Shapley值，误差可以忽略不计，这些误差由不可避免的计算误差导致；但 HarsanyiNet 的分类准确率远高于浅层ShapNet。这是因为浅层ShapNet 的表征能力有限，只能编码少数输入变量之间的交互作用。另一方面，尽管深层ShapNet 的分类准确率高于浅层ShapNet，但它无法计算精确的 Shapley值。这是因为深层ShapNet 能够编码更多输入变量之间的交互作用，但代价是无法再从理论上保证估算出的Shapley值的准确性。尽管如此，在图像分类（例如CIFAR-10 数据集上的图像分类）等更复杂的任务中，HarsanyiNet 的表现仍远优于深层ShapNet。

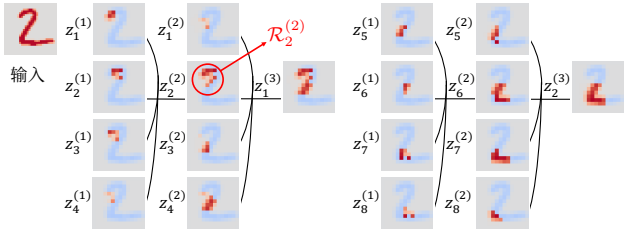


图*5.* Harsanyi 单元感受野及其对应子节点的可视化。

可视化。我们根据各方法在 MNIST 和 CIFAR-10 数据集上估算的 Shapley值生成了归因图。如图4所示[9]，，HarsanyiNet 生成的归因图几乎与通过对 20000 张抽样掩码图像进行推断而估算、且已收敛至真实Shapley值的结果相同。我们还在图5中可视化了数字图像上 Harsanyi 单元的感受野。这验证了我们可以通过合并其子节点的感受野来获得 Harsanyi 单元的感受野 $z_u^{(l)}$。

实现细节。 Harsanyi-MLP 由 3 个级联的 Harsanyi 块构成，每个块均按照公式（5）–（7）构建，且每个Harsanyi 块包含 100 个神经元。Harsanyi-CNN 在特征 $\mathbf{z}^{(0)}$上由 10 个级联的 Harsanyi 块构成，每个Harsanyi 块包含 $512 \times 16 \times 16$个神经元，其中 512 是通道数。对于使用表格数据训练的 Harsanyi-MLP，超参数设为 $\beta = 10$ 和 $\gamma = 100$；对于使用图像数据训练的Harsanyi-CNN，超参数分别设为 $\beta = 1000$ 和 $\gamma = 1$。对于 Harsanyi-MLP，我们随机选取前一层中的 10 个神经元作为初始子节点集合 $\mathcal{S}_u^{(l)}$，并将 $\boldsymbol{\tau}_u^{(l)}$中的对应维度设为 1。对于其他所有前一层的神经

[9]为便于与其他方法进行比较，对于 MNIST 数据集，Harsanyi-CNN 由 4 个级联的 Harsanyi 块构成，每个Harsanyi 块包含 $32 \times 14 \times 14$ 个神经元，其中 32 为通道数。超参数分别设为 $\beta = 100$ 和 $\gamma = 0.05$。对于 CIFAR-10 数据集，Harsanyi-CNN 由 10 个级联的 Harsanyi 块构成，每个Harsanyi 块包含 $256 \times 16 \times 16$ 个神经元，其中 256 为通道数。超参数分别设为 $\beta = 1000$ 和 $\gamma = 1$。更多详情请参见附录 F.7。

元，其在 $\boldsymbol{\tau}_u^{(l)}$中的对应维度均初始化为 -1。对于 Harsanyi-CNN，我们初始化每个参数（$\boldsymbol{\tau}_u^{(l)}$）$_i \sim \mathcal{N}$（0，0。01^2），随机选取前一层中约一半的神经元，使其满足（$\boldsymbol{\tau}_u^{(l)}$）$_i > 0$，作为初始子节点集合 $\mathcal{S}_u^{(l)}$。

关于归因评估指标的讨论。事实上，还有许多其他指标被用于评估归因方法，例如 ROAR（Hooker 等，2019年）和弱监督目标定位（Zhou 等，2016年；Schulz 等，2020年）。如表 1和图 3所示，HarsanyiNet 生成了完全准确的 Shapley值，因此，对 HarsanyiNet 生成的归因进行评估，结果应与 Shapley值相同；而先前研究已对 Shapley值的性能进行了深入分析（Lundberg & Lee，2017年；Chen 等，2019年；Wang 等，2021年；Jethani 等，2021年）。特别是，尽管 Shapley值被视为最标准的归因方法之一，并满足线性、虚设性、对称性和有效性公理，但它在所有评估指标上的表现并非总是最佳。

5. 结论

本文提出了 HarsanyiNet，可在单次前向传播中同时执行模型推理并计算输入变量的精确 Shapley值。我们从理论上证明并通过实验验证了 HarsanyiNet 计算的 Shapley值的准确性。由于不可避免的计算误差，仅产生了量级为 10^{-8}– 10^{-7} 的微小误差。此外，我们还证明 HarsanyiNet 不会限制输入变量之间的交互作用，因此具有强大的表示能力。

致谢

本研究部分得到国家自然科学基金委员会（62276165）、国家重点研发计划（2021年ZD0111602）、上海市自然科学基金（21JC1403800、21ZR1434600）、国家自然科学基金委员会（U19B2043）以及上海市科技重大项目（2021年SHZDZX0102）的资助。

参考文献

Ancona、Oztireli 和 Gross。使用多项式时间算法近似 Shapley值，以解释深度神经网络。载于 国际机器学习会议，第 272–281 页。PMLR，2019年。

Bach, S., Binder, A., Montavon, G., Klauschen, F.,M¨uller, K.-R., and Samek, W. 关于通过逐层相关性传播解释非线性分类器决策。*PLOS ONE*,10(7):e0130140,2015.

Bengio, Y.、L´eonard, N. 和 Courville, A. 用于条件计算的随机神经元梯度估计或传播。*arXiv* 预印本 *arXiv:1308.3432*，2013年。

Castro, J.、G´omez, D. 和 Tejada, J. 基于采样的 Shapley值多项式计算。《计算机与运筹研究》，36(5):1726–1730，2009年。

Chen, H.、C. Covert, I.、M. Lundberg, S. 和 Lee, S.-I. 估算 Shapley值特征归因的算法。*arXiv* 预印本 *arXiv:2207.07605*，2022年。

Chen, J.、Song, L.、Wainwright, M. J. 和 Jordan, M. I. L-Shapley 和 C-Shapley：面向结构化数据的高效模型解释。国际学习表征会议，2019年。Covert, I. 和 Lee, S.-I. 改进 KernelSHAP：使用线性回归进行实用的 Shapley值估计。载于国际人工智能与统计学会议，第 3457–3465 页。PMLR，2021年。Covert, I.、Lundberg, S. 和 Lee, S.-I. 通过移除进行解释：模型解释的统一框架。《机器学习研究杂志》，2021年。Covert, I. C.、Lundberg, S. 和 Lee, S.-I. 使用加性重要性度量理解全局特征贡献。神经信息处理系统进展，33，2020年。Deng, H.、Zou, N.、Chen, W.、Feng, G.、Du, M. 和 Hu, X. 保持互信息的反向传播：学习反演以实现忠实归因。载于第*27*届*ACM SIGKDD*知识发现与数据挖掘会议论文集，第 258–268 页，2021年。Deng, H.、Ren, Q.、Zhang, H. 和 Zhang, Q. 发现并解释深度神经网络的表征瓶颈。国际学习表征会议，2022年。Dua, D. 和 Graff, C. UCI 机器学习知识库，2017年。网址：http://archive.ics.uci.edu/ml。Fong, R. C. 和 Vedaldi, A. 通过有意义的扰动解释黑箱。载于*IEEE*国际计算机视觉会议论文集，第 3429–3437 页，2017年。Goodfellow, I. J.、Shlens, J. 和 Szegedy, C. 解释并利用对抗样本。载于国际学习表征会议，2015年。

Grabisch, M. 等人 决策中的集合函数、博弈和容量，第46卷。Springer，2016年。

Harsanyi, J. C. n人合作博弈的简化讨价还价模型。国际经济评论，4(2):194–220，1963年。

He, K., Zhang, X., Ren, S., and Sun, J. 用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，第770–778页，2016年。

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. 深度神经网络可解释性方法的基准测试。神经信息处理系统进展，32，2019年。

Jethani, N., Sudarshan, M., Covert, I. C., Lee, S.-I., and Ranganath, R. FastSHAP：实时 Shapley值估计。载于国际学习表征会议，2021年。

Jia, R., Dao, D., Wang, B., Hubis, F. A., Gurel, N. M., Li, B., Zhang, C., Spanos, C. J. 和 Song, D. 面向最近邻算法的高效任务特定数据估值。载于国际超大型数据库会议，2019年a。

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gurel, N. M., Li, B., Zhang, C., Song, D. 和 Spanos, C. 面向基于 Shapley值的高效数据估值。载于国际人工智能与统计学会议，2019年b。

Jia, R., Wu, F., Sun, X., Xu, J., Dao, D., Kailkhura, B., Zhang, C., Li, B. 和 Song, D. 可扩展性与效用：在数据重要性量化中，我们必须二者择一吗？载于*IEEE*计算机视觉与模式识别会议，2021年。

Krizhevsky, A., Hinton, G., 等. 从微小图像中学习多层特征。2009年。

LeCun, Y. 和 Cortes, C. MNIST手写数字数据库，2010年。网址：http://yann.lecun.com/ exdb/mnist/。

Li, M. 和 Zhang, Q. 神经网络真的能编码符号概念吗？国际机器学习会议，2023。

Lundberg, S. M. 和 Lee, S.-I. 一种解释模型预测的统一方法。神经信息处理系统进展，30，2017年。

Madry, A.、Makelov, A.、Schmidt, L.、Tsipras, D. 和 Vladu, A. 面向能够抵御对抗攻击的深度学习模型。载于国际学习表征会议，2018年。

Mitchell, R.、Cooper, J.、Frank, E. 和 Holmes, G. 用于估计 Shapley值的排列采样。《机器学习研究杂志》，23:1–46，2022年。Montavon, G.、Lapuschkin, S.、Binder, A.、Samek, W. 和 M¨uller, K.-R. 使用深度泰勒分解解释非线性分类决策。《模式识别》，65:211–222，2017年。Nilsback, M.-E. 和 Zisserman, A. 大规模类别自动花卉分类。载于印度计算机视觉、图形学与图像处理会议，2008年。

Okhrati, R. 和 Lipani, A. 一种用于估计 Shapley值的多线性采样算法。国际模式识别会议，2021年。

Pavlova, M., Terhljan, N., G Chung, A., Zhao, A., Surana, S., Aboutalebi, H., Gunraj, H., Sabri, A., Alaref, A. 和 Wong, A. COVID-Net CXR-2：一种增强型深度卷积神经网络设计，用于从胸部X光图像中检测COVID-19病例。发表于《*Front Med (Lausanne)*》，2022年。

Plumb, G.、Molitor, D. 和 Talwalkar, A. S. 模型无关的监督式局部解释。神经信息处理系统进展，31，2018年。

Ren, J.、Li, M.、Chen, Q.、Deng, H. 和 Zhang, Q. 定义并量化深度神经网络中稀疏概念的涌现。*IEEE*计算机视觉与模式识别会议，2023a。

Ren, Q.、Gao, J.、Shen, W. 和 Zhang, Q. 我们在证明人工智能模型中稀疏符号概念的涌现方面取得了哪些进展。*arXiv*预印本 *arXiv:2305.01939*，2023b。

Ribeiro, M. T.、Singh, S. 和 Guestrin, C. “我为什么应该相信你？”解释任意分类器的预测。载于第*22*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集，第1135–1144页，2016年。

Schulz, K., Sixt, L., Tombari, F., and Landgraf, T. 限制流动：用于归因的信息瓶颈。国际学习表征会议，2020年。

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam：通过基于梯度的定位从深度网络中获得视觉解释。发表于*IEEE*国际计算机视觉会议论文集，第618–626页，2017年。

Shapley, L. S. n人博弈的一个值。载于《博弈论的贡献》，2(28):307–317，1953。

Shrikumar, A., Greenside, P., Shcherbina, A. 和 Kundaje, A. 不只是黑箱：通过传播激活差异学习重要特征。*arXiv* 预印本 *arXiv:1605.01713*，2016年。Shrikumar, A., Greenside, P. 和 Kundaje, A. 通过传播激活差异学习重要特征。载于国际机器学习会议，第 3145–3153 页。PMLR，2017年。Simon, G. 和 Vincent, T. 一种用于估计应用于属性重要性的Shapley值的投影随机梯度算法。载于国际跨领域机器学习与知识提取会议，2020年。Simonyan, K. 和 Zisserman, A. 用于大规模图像识别的超深卷积网络。载于国际学习表征会议，2015年。Simonyan, K., Vedaldi, A. 和 Zisserman, A. 深入卷积网络：可视化图像分类模型和显著性图。国际学习表征会议，2014年。Springenberg, J. T., Dosovitskiy, A., Brox, T. 和 Riedmiller, M. 追求简洁：全卷积网络。国际学习表征会议，2015年。Strumbelj, E. 和 Kononenko, I. 运用博弈论高效解释个体分类结果。《机器学习研究杂志》，11:1–18，2010年。Sundararajan, M. 和 Najmi, A. 用于模型解释的多种 Shapley 值。载于国际机器学习会议，第 9269–9278 页。PMLR，2020年。Sundararajan, M., Taly, A. 和 Yan, Q. 深度网络的公理化归因。载于国际机器学习会议，第 3319–3328 页。PMLR，2017年。Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Izacard, G., Joulin, A., Synnaeve, G., Verbeek, J. 等。ResMLP：采用数据高效训练的图像分类前馈网络。*IEEE* 模式分析与机器智能汇刊，2022年。Wang, G., Chuang, Y.-N., Du, M., Yang, F., Zhou, Q., Tripathi, P., Cai, X. 和 Hu, X. 通过贡献型合作方选择加速 Shapley解释。载于国际机器学习会议，第 22576–22590 页。PMLR，2022年。Wang, L., Lin, Z. Q. 和 Wong, A. COVID-Net：一种定制的深度卷积神经网络设计，用于检测胸部X射线图像中的 COVID-19 病例。载于《科学报告》，2020年。Wang, R.、Wang, X. 和 Inouye, D. Shapley解释网络。载于国际学习表征会议，2021年。Weber, R. J. 博弈的概率值。载于《*Shapley*值：致敬*Lloyd S. Shapley*的论文集》，101–119，1988年。Wightman, R.、Touvron, H. 和 J´egou, H. ResNet 反击：timm中改进的训练流程。载于神经信息处理系统，2021年。Young, H. P. 合作博弈的单调解。国际博弈论期刊，14:65–72，1985年。Zhou, B.、Khosla, A.、Lapedriza, A.、Oliva, A. 和 Torralba, A. 学习用于判别性定位的深度特征。载于*IEEE*计算机视觉与模式识别会议论文集，第2921–2929页，2016年。Zintgraf, L. M.、Cohen, T. S.、Adel, T. 和 Welling, M. 可视化深度神经网络的决策：预测差异分析。国际学习表征会议，2017年。

A. Shapley值

本节将重新审视Shapley值所满足的四条公理，这些公理确保Shapley值能够提供相对忠实的归因值。考虑如下合作博弈 V： $2^N \mapsto \mathbb{R}$，其中一组 n 参与者$N = \{1, 2, \ldots, n\}$ 进行合作并赢得奖励 R。这里， $V(S)$等价于论文中提到的 $v(\mathbf{x}_S) - v(\mathbf{x}_\emptyset)$，并且有 $V(\emptyset) = 0$。Young（1985）证明，Shapley值是满足四条公理的唯一解，包括线性公理、虚设性公理、对称性公理和有效性公理（Weber，1988）。

（1）线性公理：如果对于所有 $S \subseteq N$，博弈 $V(\cdot)$ 是两个博弈 $U(\cdot)$、 $W(\cdot)$ 的线性组合，即 $V(S) = U(S) + W(S)$ 和 $(c \cdot V)(S) = c \cdot V(S)\ \forall c \in \mathbb{R}$，那么博弈 V 中的Shapley值也是博弈 U和 W中Shapley值的线性组合，即 $\forall i \in N$， $\phi^V(i) = \phi^U(i) + \phi^W(i)$ 以及 $\phi^{c \cdot V}(i) = c \cdot \phi^V(i)$。

（2）虚设公理：如果对于每个 $S \subseteq N \setminus \{i\}$ 都有 $V(S \cup \{i\}) = V(S) + V(\{i\})$，则参与者 i 被定义为虚设参与者。虚设参与者 i 满足 $\phi(i) = V(\{i\})$，这表明参与者 i 单独影响总奖励，而不会在 N 中与其他参与者互动或合作。

(3) 对称公理： 对于两个参与者 i和 j，如果 $\forall S \subseteq N \setminus \{i, j\}, V(S \cup \{i\}) = V(S \cup \{j\})$，则参与者 i 和 j 的Shapley值相等，即 $\phi(i) = \phi(j)$。

(4)效率公理：该 总奖励等于每个参与者的 Shapley值之和，即 $\sum_{i=1}^{n} \phi(i) = V(N)$.

B. 定理证明

本节将证明论文中的定理。

定理2。 设网络输出 $v(\mathbf{x}) \in \mathbb{R}$ 表示为 $v(\mathbf{x}) = \sum_{l=1}^{L} (\mathbf{w}_v^{(l)})^\intercal \mathbf{z}^{(l)}(\mathbf{x})$，根据公式 (2)。这样，在网络输出 $v(\mathbf{x})$ 上计算的集合 S 中输入变量之间的 ***Harsanyi*交互**可以表示为 $I(S) = \sum_{l=1}^{L} \sum_{u=1}^{m^{(l)}} w_u^{(l)} J_u^{(l)}(S)$。

证明。我们有

$$\begin{aligned} v(\mathbf{x}) &= \sum\nolimits_{l=1}^{L} (\mathbf{w}_v^{(l)})^\intercal \mathbf{z}^{(l)}(\mathbf{x}) \\ &= \sum\nolimits_{l=1}^{L} \sum\nolimits_{u=1}^{m^{(l)}} w_u^{(l)} z_u^{(l)}(\mathbf{x}). \end{aligned}$$

根据 Harsanyi交互的线性性质，如果 $\forall S \subseteq N$, $v(\mathbf{x}_S) = u(\mathbf{x}_S) + w(\mathbf{x}_S)$ 和 $(cv)(\mathbf{x}_S) = c \cdot v(\mathbf{x}_S)\ \forall c \in \mathbb{R}$，那么 Harsanyi交互 $I^v(S)$ 也是 $I^u(S)$ 和 $I^w(S)$ 的线性组合，即 $\forall S \subseteq N, I^v(S) = I^u(S) + I^w(S)$ 和 $I^{(cv)}(S) = c \cdot I^v(S)$。因此，由于 $J_u^{(l)}(S)$ 表示在函数 $z_u^{(l)}(\mathbf{x})$ 上计算得到的 Harsanyi交互，我们有，在网络输出 $v(\mathbf{x})$ 上计算得到的 Harsanyi交互是 $J_u^{(l)}(S)$ 的线性组合，即

$$I(S) = \sum\nolimits_{l=1}^{L} \sum\nolimits_{u=1}^{m^{(l)}} w_u^{(l)} J_u^{(l)}(S).$$

□

定理 3 （从中间层的 Harsanyi 单元推导 Shapley值）。 ***Shapley*值** $\phi(i)$ 可计算为

$$\phi(i) = \sum\nolimits_{l=1}^{L} \sum\nolimits_{u=1}^{m^{(l)}} \frac{1}{|\mathcal{R}_u^{(l)}|} w_u^{(l)} z_u^{(l)}(\mathbf{x}) \mathbb{1}(\mathcal{R}_u^{(l)} \ni i).$$

证明。 根据定理1和定理2，我们有

$$\begin{aligned}\phi(i) &= \sum\nolimits_{S\subseteq N: S\ni i} \frac{1}{|S|} I(S) \\ &= \sum\nolimits_{S\subseteq N} \frac{1}{|S|} I(S)\mathbb{1}(S \ni i) \\ &= \sum_{l=1}^{L}\sum_{u=1}^{m^{(l)}} \frac{1}{|\mathcal{R}_u^{(l)}|} w_u^{(l)} z_u^{(l)}(\mathbf{x})\mathbb{1}(\mathcal{R}_u^{(l)} \ni i).\end{aligned}$$

□

定理4. 根据公式(5)–(7)，神经元 $z_u^{(l)}$的感受野 $\mathcal{R}_u^{(l)}$会自动满足要求*1*和要求*2*。神经元(l, u) 的感受野 $\mathcal{R}_u^{(l)}$递归定义为 $\mathcal{R}_u^{(l)} := \cup_{(l',u')\in\mathcal{S}_u^{(l)}} \mathcal{R}^{(l'}$) u' , 使得 $\mathcal{R}_u^{(1)} := \mathcal{S}_u^{(1)}$.

证明. **(1) 证明神经元 $z_u^{(l)}$的感受野 $\mathcal{R}_u^{(l)}$满足要求1** .

给定任意两个样本 $\tilde{\mathbf{x}} = \tilde{\mathbf{z}}^{(0)}$ 和$\mathbf{x} = \mathbf{z}^{(0)}$，为满足要求1，我们将证明：如果 $\forall\, i \in \mathcal{R}_u^{(l)}, \tilde{\mathbf{x}}_i = \mathbf{x}_i$，则 $z_u^{(l)}(\tilde{\mathbf{x}}) = z_u^{(l)}$(x</style)。

首先，对于第一层，$\forall u'$，$\mathcal{R}_{u'}^{(1)} = \mathcal{S}_{u'}^{(1)} \subseteq \mathcal{R}_u^{(l)}$，我们证明 $z_{u'}^{(1)}(\tilde{\mathbf{x}}) = z_{u'}^{(1)}(\mathbf{x})$。

我们得到 $g_{u'}^{(1)}(\tilde{\mathbf{x}}) = (\mathbf{A}_{u'}^{(1)})^\intercal \cdot \left(\mathbf{\Sigma}_{u'}^{(1)} \cdot \tilde{\mathbf{z}}^{(0)}\right) = (\mathbf{A}_{u'}^{(1)})^\intercal \cdot \boldsymbol{\zeta}$，其中 $\forall i \in \mathcal{R}_{u'}^{(1)} = \mathcal{S}_{u'}^{(1)} \subseteq \mathcal{R}_u^{(l)}, \boldsymbol{\zeta}_i = \tilde{\mathbf{z}}_i^{(0)}$，否则 $\boldsymbol{\zeta}_i = 0$。

我们还得到 $g_{u'}^{(1)}(\mathbf{x}) = (\mathbf{A}_{u'}^{(1)})^\intercal \cdot \left(\mathbf{\Sigma}_{u'}^{(1)} \cdot \mathbf{z}^{(0)}\right) = (\mathbf{A}_{u'}^{(1)})^\intercal \cdot \boldsymbol{\eta}$，其中 $\forall i \in \mathcal{R}_{u'}^{(1)}, \boldsymbol{\eta}_i = \mathbf{z}_i^{(0)} = \tilde{\mathbf{z}}_i^{(0)}$，否则 $\boldsymbol{\eta}_i = 0$。

因此，$g_{u'}^{(1)}(\tilde{\mathbf{x}}) = g_{u'}^{(1)}(\mathbf{x})$ 且 $z_{u'}^{(1)}(\tilde{\mathbf{x}}) = z_{u'}^{(1)}(\mathbf{x})$。

其次，我们利用上述结论证明 $z_u^{(l)}(\tilde{\mathbf{x}}) = z_u^{(l)}(\mathbf{x})$。

对于第二层，$\forall u', \mathcal{R}_{u'}^{(2)} = \cup_{(1,u'')\in\mathcal{S}_{u'}^{(2)}} \mathcal{R}_{u''}^{(1)} \subseteq \mathcal{R}_u^{(l)}$，我们可以轻松得到 $z_{u'}^{(2)}(\tilde{\mathbf{x}}) = z_{u'}^{(2)}(\mathbf{x})$，因为其子节点是从 $\forall u'', \mathcal{R}_{u''}^{(1)} = \mathcal{S}_{u''}^{(1)} \subseteq \mathcal{R}_u^{(l)}$ 中选取的，并且其输出满足 $z_{u''}^{(1)}(\tilde{\mathbf{x}}) = z_{u''}^{(1)}(\mathbf{x})$。类似地，我们可以递归地推导出$z_u^{(l)}(\tilde{\mathbf{x}}) = z_u^{(l)}(\mathbf{x})$。

这样，我们证明了神经元 $z_u^{(l)}$的感受野 $\mathcal{R}_u^{(l)}$满足要求1。

(2) 证明神经元 $z_u^{(l)}$的感受野 $\mathcal{R}_u^{(l)}$满足要求2。

给定样本 $\mathbf{x} = \mathbf{z}^{(0)}$ 及其任意掩码样本$\mathbf{x}_S = \mathbf{z}_S^{(0)}$，为满足要求2，我们将证明$z_u^{(l)}(\mathbf{x}_S) = z_u^{(l)}(\mathbf{x}) \cdot \prod_{i\in\mathcal{R}_u^{(l)}} \mathbb{1}(i \in S)$。具体而言，我们将分别证明，在条件 **(1)** $\forall S \supseteq \mathcal{R}_u^{(l)}$、**(2)**$\forall S \subsetneq \mathcal{R}_u^{(l)}$，或 $\forall S, S \cup \mathcal{R}_u^{(l)} \neq S$和 $S \cup \mathcal{R}_u^{(l)} \neq \mathcal{R}_u^{(l)}$ 下，可以得到 $z_u^{(l)}(\mathbf{x}_S) = z_u^{(l)}(\mathbf{x}) \cdot \prod_{i\in\mathcal{R}_u^{(l)}} \mathbb{1}(i \in S)$。

首先，我们可以轻松得到 $\forall S \supseteq \mathcal{R}_u^{(l)}$，$z_u^{(l)}(\mathbf{x}_S) = z_u^{(l)}(\mathbf{x}) \cdot \prod_{i\in\mathcal{R}_u^{(l)}} \mathbb{1}(i \in S)$。由于 $\forall i \in \mathcal{R}_u^{(l)}$，$(\mathbf{x}_S)_i = \mathbf{x}_i$，让我们使用（1）中已证明的结论来推导 $z_u^{(l)}(\mathbf{x}_S) = z_u^{(l)}(\mathbf{x}) = z_u^{(l)}(\mathbf{x}) \cdot \prod_{i\in\mathcal{R}_u^{(l)}} \mathbb{1}(i \in S)$。

其次，我们证明，在 $\forall S \subsetneq \mathcal{R}_u^{(l)}$ 或 $\forall S$，$S \cup \mathcal{R}_u^{(l)} \neq S$ 以及 $S \cup \mathcal{R}_u^{(l)} \neq \mathcal{R}_u^{(l)}$ 的条件下，可以得到$z_u^{(l)}(\mathbf{x}_S) = z_u^{(l)}(\mathbf{x}) \cdot \prod_{i\in\mathcal{R}_u^{(l)}} \mathbb{1}(i \in S)$。

令 $\mathbf{x}_S$ 表示在样本 $\mathbf{x}$中掩蔽集合 $N \setminus S$ 内由 $\mathbf{b}$ 指定的变量后得到的样本，则 $\mathbf{z}^{(0)} = \mathbf{x} - \mathbf{b} \in \mathbb{R}^n$。在这两种设定中，至少存在一个变量 j，它属于 $\mathcal{R}_u^{(l)}$但不属于 S，即 $\exists j \in \mathcal{R}_u^{(l)}$，$j \notin S$，我们有 $(\mathbf{x}_S)_j = b$、$(\mathbf{z}_S^{(0)})_j = 0$以及 $\prod_{i\in\mathcal{R}_u^{(l)}} \mathbb{1}(i \in S) = 0$。

对于第一层，至少存在一个神经元 $(1, u')$ 满足 $j \in \mathcal{R}_{u'}^{(1)} = \mathcal{S}_{u'}^{(1)} \subseteq \mathcal{R}_u^{(l)}$。那么 $\forall u'$，$h_{u'}^{(1)}(\mathbf{x}_S) = g_{u'}^{(1)}(\mathbf{x}_S) \cdot \prod_{(0,\, u'')\in\mathcal{S}_{u'}^{(1)}} \mathbb{1}(z_{u''}^{(0)}(\mathbf{x}_S) \neq 0) = g_{u'}^{(1)}(\mathbf{x}_S) \cdot \mathbb{1}((\mathbf{z}_S^{(0)})_j \neq 0) = 0$ 和 $z_{u'}^{(1)}(\mathbf{x}_S) = 0$。由于 $j \in \mathcal{R}_u^{(l)} =$

$\cup_{(l',u'')\in\mathcal{S}_u^{(l)}}\mathcal{R}^{(l'}{}_{)\,u''}$，至少存在一个神经元 $(l,\ u')$ 会递归地影响神经元 $(l,\ u)$，即，$h_u^{(l)}(\mathbf{x}_S)=0$ 和 $z_u^{(l)}(\mathbf{x}_S)=0$。因此，$z_u^{(l)}(\mathbf{x}_S)=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in S)=0$。

由此，我们证明了神经元 $z_u^{(l)}$ 的感受野 $\mathcal{R}_u^{(l)}$ 满足要求2。

□

C. 引理1的证明

引理1 (Harsanyi单元的Harsanyi交互)**.** 令 *Harsanyi* 单元 $z_u^{(l)}(\mathbf{x})$ 的输出作为奖励。接着，令 $J_u^{(l)}(S)$ 表示关于函数 $z_u^{(l)}(\mathbf{x})$ 的 *Harsanyi*交互。根据要求*1*和要求*2*，我们有 $J_u^{(l)}(\mathcal{R}_u^{(l)})=z_u^{(l)}(\mathbf{x})$，且$\forall S\neq\mathcal{R}_u^{(l)}, J_u^{(l)}(S)=0$。

证明。 根据定义2，即*i.e.*，$I(S)=v(S)-\sum_{L\subsetneq S}I(L)$满足 $I(\emptyset):=0$，以及要求2，即*i.e.*，$z_u^{(l)}(\mathbf{x}_S)=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in S)$，Harsanyi 单元的 Harsanyi交互可写为：

$$\begin{aligned}J_u^{(l)}(S)&=z_u^{(l)}(\mathbf{x}_S)-\sum_{L\subsetneq S}J_u^{(l)}(L)\\&=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in S)-\sum_{L\subsetneq S}J_u^{(l)}(L)\end{aligned}$$

(1) $J_u^{(l)}(\mathcal{R}_u^{(l)})=z_u^{(l)}(\mathbf{x})$。

首先，我们使用归纳法证明 $\forall L\subsetneq\mathcal{R}_u^{(l)},\ J_u^{(l)}(L)=0$。
如果 $|L|=1,\ \forall L'\subseteq L\subsetneq\mathcal{R}_u^{(l)}$，我们有 $\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in L')=0$，那么可得 $J_u^{(l)}(L')=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in L')=0$。

假设如果 $|L|=k,\forall L'\subseteq L\subsetneq\mathcal{R}_u^{(l)}$，我们有 $J_u^{(l)}(L')=0$。
那么如果 $|L|=k+1,\forall L'\subseteq L\subsetneq\mathcal{R}_u^{(l)}$，我们有 $\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in L)=0$，并且 $\forall L'\subsetneq L,\ J_u^{(l)}(L')=0$。因此，可得$J_u^{(l)}(L)=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in L)-\sum_{L'\subsetneq L}J_u^{(l)}(L')=0$。

由此，我们证明了 $\forall 1\leq|L|<|\mathcal{R}_u^{(l)}|,\forall L\subsetneq\mathcal{R}_u^{(l)},J_u^{(l)}(L)=0$。

其次，我们利用已证明的结论 $\forall L\subsetneq\mathcal{R}_u^{(l)},J_u^{(l)}(L)=0$，推导出 $J_u^{(l)}(\mathcal{R}_u^{(l)})=z_u^{(l)}(\mathbf{x})$。
由于$\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in\mathcal{R}_u^{(l)})=1$，我们得到 $J_u^{(l)}(\mathcal{R}_u^{(l)})=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in\mathcal{R}_u^{(l)})-\sum_{L\subsetneq\mathcal{R}_u^{(l)}}J_u^{(l)}(L)=z_u^{(l)}(\mathbf{x})$。由此，我们证明了 $J_u^{(l)}(\mathcal{R}_u^{(l)})=z_u^{(l)}(\mathbf{x})$。

(2) $\forall S\neq\mathcal{R}_u^{(l)},\ J_u^{(l)}(S)=0$。

为了证明 $\forall S\neq\mathcal{R}_u^{(l)},\ J_u^{(l)}(S)=0$，我们将分别证明，在满足条件 **(1)** $\forall S\subsetneq\mathcal{R}_u^{(l)}$、**(2)** $\forall S\supsetneq\mathcal{R}_u^{(l)}$ 和 **(3)**$\forall S,\ S\cup\mathcal{R}_u^{(l)}\neq S$ 以及 $S\cup\mathcal{R}_u^{(l)}\neq\mathcal{R}_u^{(l)}$ 时，可以得到 $J_u^{(l)}(S)=0$。

首先，我们已经证明了 $\forall S\subsetneq\mathcal{R}_u^{(l)},\ J_u^{(l)}(S)=0$。

其次，我们用归纳法证明 $\forall S\supsetneq\mathcal{R}_u^{(l)},\ J_u^{(l)}(S)=0$。
在此设定下，$\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in S)=1$。若 $|S|=|\mathcal{R}_u^{(l)}|+1,\forall S\supsetneq\mathcal{R}_u^{(l)}$，则有 $J_u^{(l)}(S)=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in S)-\sum_{L\subsetneq S}J_u^{(l)}(L)=z_u^{(l)}(\mathbf{x})-[J_u^{(l)}(\mathcal{R}_u^{(l)})+\sum_{L\subsetneq S,\ L\neq\mathcal{R}_u^{(l)}}J_u^{(l)}(L)]=0$。（同样，$\forall L\subsetneq S$和 $L\neq\mathcal{R}_u^{(l)},\ J_u^{(l)}(L)=0$也可以通过归纳法证明。）

假设当 $|S|=|\mathcal{R}_u^{(l)}|+k,\ \forall S\supsetneq\mathcal{R}_u^{(l)}$时，有 $J_u^{(l)}(S)=0$。

因此，如果 $|S|=|\mathcal{R}_u^{(l)}|+(k+1)\ \forall S\supsetneq\mathcal{R}_u^{(l)}$，则有 $J_u^{(l)}(S)=0$。

第三 让我们使用归纳法证明 $\forall S,\ S\cup\mathcal{R}_u^{(l)}\neq S$ 和 $S\cup\mathcal{R}_u^{(l)}\neq\mathcal{R}_u^{(l)},\ J_u^{(l)}(S)=0$ 在此设定下，$\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in S)=0$。于是我们有 $J_u^{(l)}(S)=z_u^{(l)}(\mathbf{x})\cdot\prod_{i\in\mathcal{R}_u^{(l)}}\mathbb{1}(i\in S)-\sum_{L\subsetneq S}J_u^{(l)}(L)=0$。（类似地，$\forall L\subsetneq S,\ J_u^{(l)}(S)=0$ 可通过归纳法证明。）

通过这种方式，我们证明了 $\forall S\neq\mathcal{R}_u^{(l)},\ J_u^{(l)}(S)=0$。

□

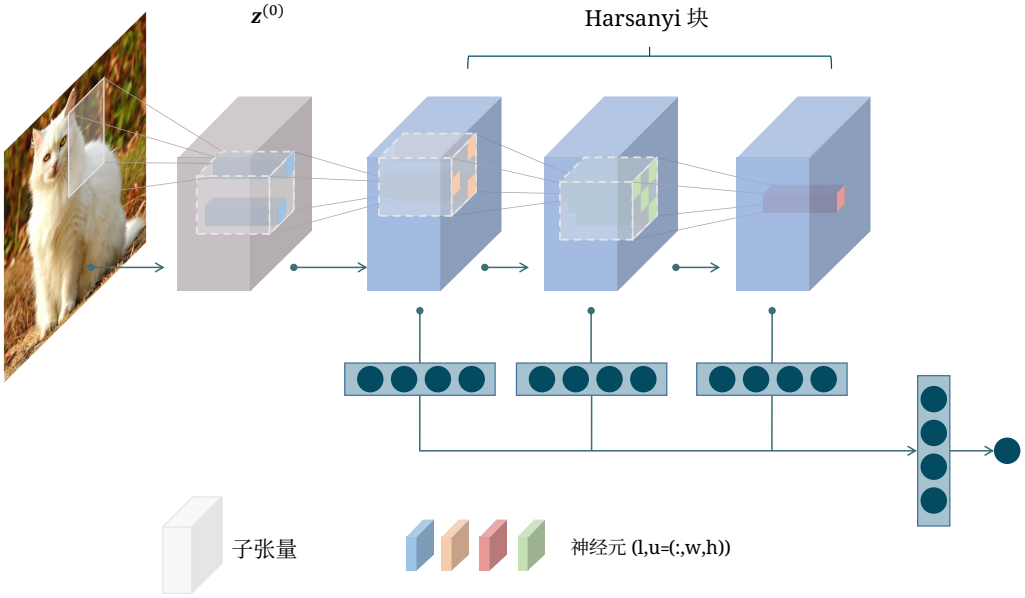
D. 关于公式（5）–（7）的讨论

第3.3节介绍了，Harsanyi 块中神经元 $(l,\ u)$ 的神经激活 $z_u^{(l)}(\mathbf{x})$ 是通过依次执行线性运算（公式（5））、与运算（公式（6））和 ReLU 操作（公式（7））计算得出的。下面我们将进一步讨论上述三种运算。

与传统深度神经网络中的线性层不同，公式（5）表明，在所有先前的 $(l-1)$ 个块中的神经元里，只有子节点 $\mathbf{\Sigma}_u^{(l)}\cdot\mathbb{z}^{(l-1)}$ 的输出会影响神经元 $(l,\ u)$ 的输出。公式（6）表示，如果 $\mathcal{S}_u^{(l)}$ 中的所有子节点都被激活，那么激活得分 $g_u^{(l)}(\mathbf{x})$ 就能通过与运算，即 $h_u^{(l)}(\mathbf{x})=g_u^{(l)}(\mathbf{x})$。反之，如果任一子节点未被激活，即 $\exists(l',\ u')\in\mathcal{S}_u^{(l)}$，例如 $z^{(l')}{}_{u'}(\mathbf{x})=0$，那么我们就有 $h_u^{(l)}(\mathbf{x})=0$。

E. Harsanyi-CNN 的证明与实现细节

Harsanyi-CNN 架构。图6展示了 Harsanyi-CNN 的架构。如第4.1节所述，我们首先应用卷积层、最大池化层和 ReLU 层，得到特征 $\mathbf{z}^{(0)}$。然后，我们在 $\mathbf{z}^{(0)}$ 上构建级联的 Harsanyi 块。与传统卷积神经网络类似，每个Harsanyi块中卷积层的每个神经元 $(l,u=(c,w,h))$ 都对应于前一层中的一个子张量 $\mathbf{T}_u^{(l)}\in\mathbb{R}^{C\times K\times K}$，其中 C 是前一层的通道数，$K\times K$ 是二维卷积核大小。位于相同位置但处于不同通道的神经元 $(l,u=(:,w,h))$ 对应于同一个子张量 $\mathbf{T}_u^{(l)}$。每个神经元 (l,u) 的子节点集合 $\mathcal{S}_u^{(l)}$从子张量 $\mathbf{T}_u^{(l)}$中选取。此外，位于相同位置但处于不同通道的神经元 $(l',u'=(:,w',h'))$ 同时属于子节点集合 $\mathcal{S}_u^{(l)}$。Harsanyi块的输出按照公式（5）–（7）构建。最后，网络输出 $v(\mathbf{x})$ 的每个维度都通过线性变换和跳跃连接，构建为各个 Harsanyi块输出的加权和。



*Figu*图 *6.* Harsanyi-CNN 架构示意图　　结构。

设定2中结论的证明：基于如下设计，即让所有神经元（$l, u=(1, h, w)) \ldots, (l, u=(C, h, w))$共享相同的参数 $\boldsymbol{\tau}_u^{(l)}$，不同通道（$c=1, \ldots, C$）上相同位置（$h, w$）的所有 Harsanyi 单元（$l, u=(c, h, w)$）具有相同的感受野 $\mathcal{R}_{u=(:, h, w)}^{(l)}$，并且共同贡献于同一个 Harsanyi交互 $I(S=\mathcal{R}_{u=(:, h, w)}^{(l)})$。

证明。根据实现 $\forall(l, u)\ \boldsymbol{\tau}_{u=(1, h, w)}^{(l)}=\boldsymbol{\tau}_{u=(2, h, w)}^{(l)}=\cdots=\boldsymbol{\tau}_{u=(C, h, w)}^{(l)} \in \mathbb{R}^{CK^2}$，我们将证明 $\forall(l, u)\ \mathcal{R}_{u=(1, h, w)}^{(l)}=\mathcal{R}_{u=(2, h, w)}^{(l)}=\cdots=\mathcal{R}_{u=(C, h, w)}^{(l)}$。

由于 $\forall(l, u), (\boldsymbol{\Sigma}_u^{(l)})_{i,i}=\mathbb{1}((\boldsymbol{\tau}_{(l)u})_i>0)$，对于任意二值对角矩阵 $\boldsymbol{\Sigma}_u^{(l)}$，我们有 $\forall(l, u)\ \boldsymbol{\Sigma}_{u=(1, h, w)}^{(l)}=\boldsymbol{\Sigma}_{u=(2, h, w)}^{(l)}=\cdots=\boldsymbol{\Sigma}_{u=(C, h, w)}^{(l)}$。子节点集合 $\mathcal{S}_u^{(l)}$由 $\boldsymbol{\Sigma}_u^{(l)}$实现，因此我们有 $\forall(l, u)\ \mathcal{S}_{u=(1, h, w)}^{(l)}=\mathcal{S}_{u=(2, h, w)}^{(l)}=\cdots=\mathcal{S}_{u=(C, h, w)}^{(l)}$。根据公式（8），我们递归地从 $\mathcal{S}_u^{(l)}$推导出 $\mathcal{R}_u^{(l)}$，因此我们有$\forall(l, u)\ \mathcal{R}_{u=(1, h, w)}^{(l)}=\mathcal{R}_{u=(2, h, w)}^{(l)}=\cdots=\mathcal{R}_{u=(C, h, w)}^{(l)}$。这样，不同通道（$c=1, \ldots, C$）上相同位置（$h, w$）的 Harsanyi 单元（$l, u=(c, h, w)$）具有相同的感受野 $\mathcal{R}_{u=(:, h, w)}^{(l)}$。

接下来，我们将说明，无论是将 C 个通道视为 C 个 Harsanyi 单元，还是将 C 个通道合并视为一个 Harsanyi 单元，两种情况下它们的 Harsanyi 交互作用都相同。

将 C 个通道视为 C 个 Harsanyi 单元时，第 l 层总共有 $m^{(l)}=H\times W\times C$ 个 Harsanyi 单元。我们有$I(S=\mathcal{R}_{u=(1,\ h,\ w)}^{(l)})=I(S=\mathcal{R}_{u=(2,\ h,\ w)}^{(l)})=\cdots=I(S=\mathcal{R}_{u=(C,\ h,\ w)}^{(l)})$，简写为 $I(S=\mathcal{R}_u^{(l)})$。根据定理2和引理1，我们有

$$I(S=\mathcal{R}_u^{(l)})=\sum\nolimits_{l=1}^{L}\sum\nolimits_{u=1}^{m^{(l)}} w_{v,u}^{(l)} J_u^{(l)}(S=\mathcal{R}_u^{(l)})=\sum\nolimits_{l=1}^{L}\sum\nolimits_{u=1}^{H\times W\times C} w_{v,u}^{(l)} z_u^{(l)}(\mathbf{x})$$

其中 $w_v^{(l)}{}_{,u}\in\mathbb{R}$和 $z_u^{(l)}(\mathbf{x})\in\mathbb{R}$。根据公式（5）–（7），注意到 $\forall c\in\{1, 2, \ldots, C\}$、$(l、\ u=(c、\ h、\ w))$ 共享相同的子节点 $\mathcal{S}_{u=(1、\ h、\ w)}^{(l)}=\mathcal{S}_{u=(2、\ h、\ w)}^{(l)}=\cdots=\mathcal{S}_{u=(C、\ h、\ w)}^{(l)}$，因此，不同通道上位于相同位置的 $h_{u=(c、\ h、\ w)}^{(l)}(\mathbf{x})$ 会因子节点上的与运算而同时激活或失活。此外，$z_u^{(l)}(\mathbf{x})$ 由子节点 $g_u^{(l)}(\mathbf{x})=(\mathbf{A}_u^{(l)})^\intercal\cdot\left(\boldsymbol{\Sigma}_u^{(l)}\cdot\mathbb{z}^{(l-1)}\right)$ 的线性组合决定，其中 $\mathbf{A}_u^{(l)}\in\mathbb{R}^{CK^2}$ 是卷积核的参数（共有 C 个卷积核，记为 $\mathbf{B}_u^{(l)}\in\mathbb{R}^{(CK^2)}$），可在不同通道的相同位置生成 C 个 Harsanyi 单元）；$\boldsymbol{\Sigma}_u^{(l)}\in\mathbb{R}^{(CK^2)\times(CK^2)}$ 表示选定的子节点，而 $\mathbf{z}^{(l-1)}\in\mathbb{R}^{CK^2}$ 表示位于卷积核覆盖范围内的第 $l-1$ 层特征图。

将 C 个通道视为一个 Harsanyi 单元后，l层总共有 $m^{(l)}=H\times W$ 个 Harsanyi 单元。我们用 $I(S=\mathcal{R}_{u=(:\ h,w)}^{(l)})$ 表示这种情况。根据定理2和引理1，我们有

$$I(S=\mathcal{R}_{u=(:,h,w)}^{(l)})=\sum\nolimits_{l=1}^{L}\sum\nolimits_{u=1}^{m^{(l)}} w_{v,u}^{(l)} J_u^{(l)}(S=\mathcal{R}_u^{(l)})=\sum\nolimits_{l=1}^{L}\sum\nolimits_{u=1}^{H\times W}(\mathbf{w}_{v,u}^{(l)})^\intercal\mathbf{z}_u^{(l)}(\mathbf{x})$$

其中，$\mathbf{w}_v^{(l)}{}_{,u}\in\mathbb{R}^C$和 $\mathbf{z}_u^{(l)}(\mathbf{x})\in\mathbb{R}^C$。根据公式（5）–（7），注意到 $\forall c\in\{1, 2, \ldots, C\}, \mathcal{S}_{u=(:\ h,w)}^{(l)}=\mathcal{S}_{u=(c,h,w)}^{(l)}$，那么这个单个 C维 Harsanyi 单元与上述情况中的 C个 Harsanyi 单元具有相同的激活状态。此外，$\mathbf{z}_u^{(l)}(\mathbf{x})$由子节点 $\mathbf{g}_u^{(l)}(\mathbf{x})=(\mathbf{B}_u^{(l)})^\intercal\cdot\left(\boldsymbol{\Sigma}_u^{(l)}\cdot\mathbb{z}^{(l-1)}\right)\in\mathbb{R}^C$的线性组合决定，其中 $\mathbf{B}_u^{(l)}\in\mathbb{R}^{(CK^2)\times C}$ 是总共 C 个卷积核的参数，$\boldsymbol{\Sigma}_u^{(l)}\in\mathbb{R}^{(CK^2)\times(CK^2)}$表示选定的子节点，$\mathbf{z}^{(l-1)}\in\mathbb{R}^{CK^2}$ 表示卷积核覆盖范围内第 $(l-1)$ 层的特征图。

通过这种方式，我们证明了不同通道（$c=1, \ldots, C$）上相同位置（h, w）的 Harsanyi 单元（$l, u=(c, h, w)$）具有相同的感受野 $\mathcal{R}_{u=(:\ h,w)}^{(l)}$，并且共同贡献于相同的 Harsanyi交互 $I(S=\mathcal{R}_{u=(:\ h,w)}^{(l)})$。

□

F. 更多实验结果与细节

F.1. Harsanyi 网络在更具挑战性数据集上的实验结果

为进一步探究 Harsanyi 网络的分类性能，我们在更具挑战性的数据集上进行了实验，包括 Oxford Flowers-102（Nilsback 和 Zisserman，2008年）和 COVIDx 数据集（Wang 等人，2020年）。为了

表 *3.* HarsanyiNet 与基线模型在更具挑战性的数据集上的分类准确率（%） s

数据集	HarsanyiNet	基线模型
Oxford Flowers-102	95.48	97.9（ResNet50（Wightman等人，2021年））
COVIDx	96.75	96.3（COVID-Net CXR-2（Pavlova 等人，2022年））

表 *4.*.....之间的误差 HarsanyiNet 计算出的 Shapley值与采样方法估算出的 Shapley值

数据集	Shapley值误差（5000 次迭代）	Shapley值误差（10000 次迭代）
MNIST	0.017	0.012
CIFAR-10	0.007	0.004

比较 HarsanyiNet 与传统深度神经网络的分类准确率，我们使用 ResNet-50（He 等人，2016年）和 COVID-Net CXR-2（Pavlova 等人，2022年）作为基线模型，并将结果列于表 3 中。具体而言，我们使用预训练 VGG-16 模型（Simonyan 和 Zisserman，2015年）的大小为 $512 \times 14 \times 14$ 的中间层特征作为 $\mathbf{z}^{(0)}$，随后使用第 4 节所述的相同超参数，在 $\mathbf{z}^{(0)}$ 上训练 HarsanyiNet。

如表 3 所示，HarsanyiNet 在 Oxford Flowers-102 上的分类准确率略低于 ResNet-50。然而，在医学数据集 COVIDx 上，HarsanyiNet 的分类准确率略高于 COVID-Net CXR-2。尽管在某些数据集上分类准确率略有牺牲，HarsanyiNet 仍能通过一次前向传播计算出精确的 Shapley值，这是它相较于其他神经网络的主要优势。

F.2. 用于验证 Harsanyi 网络上 Shapley值准确性的实验结果

为进一步验证 HarsanyiNet 在高维图像数据集上计算 Shapley值的准确性，我们将 HarsanyiNet 计算出的 Shapley值与采样方法估算出的 Shapley值进行了比较。具体而言，我们分别在 MNIST 数据集和 CIFAR-10 数据集上运行采样算法，迭代 5000 次和 10000 次。表 4 展示了 HarsanyiNet 计算出的 Shapley值与采样算法估算出的 Shapley值之间的均方根误差（RMSE）。两种方法的估计误差都很小。不过，我们需要强调的是，尽管采样次数较多时采样算法更加准确，但估算出的 Shapley值与真实夏普利值之间仍存在不可忽略的误差。

F.3. Harsanyi 网络训练成本的实验结果

为进一步探究 Harsanyi 网络的训练成本，我们在 Census、MNIST 和 CIFAR-10 数据集上进行了实验，以评估 Harsanyi 网络与规模相当的传统深度神经网络的训练成本。表 5 显示，训练 HarsanyiNet 的计算成本高于训练可比深度神经网络；对于参数数量相同的情况，HarsanyiNet 的计算成本约为传统深度神经网络的两倍。

F.4. Harsanyi 网络鲁棒性实验结果

我们进行了更多实验，以分析 HarsanyiNet 的鲁棒性。具体而言，我们评估了分类性能的对抗鲁棒性，以及估计得到的 Shapley值的对抗鲁棒性（Jia 等人，2019b）。

为评估 HarsanyiNet 分类性能的对抗鲁棒性，我们在 CIFAR-10 数据集上进行了实验，通过考察模型在对抗样本测试集上的分类准确率来评估其模型鲁棒性。为了生成对抗样本，我们使用了 FGSM 攻击（Goodfellow 等，2015），这是一种基于梯度的方法，最大扰动幅度为 8/255（Madry 等，2018）。表 6 显示，HarsanyiNet 对抗样本的分类准确率略高于 ResNet-18（He 等，2016）。

为估计 HarsanyiNet 上估计所得 Shapley值的对抗鲁棒性，我们通过计算对抗样本与自然样本之间 Shapley值差异的 ℓ_2 范数，评估其估计所得 Shapley值的鲁棒性，即，$||\boldsymbol{\phi}^{nat} - \boldsymbol{\phi}^{adv}||_{\ell_2}$，其中 $\boldsymbol{\phi}^{nat}$ 表示自然样本的 Shapley值，$\boldsymbol{\phi}^{adv}$ 表示对抗样本的 Shapley值。为计算 ResNet-18 模型的 Shapley值，我们使用采样算法进行 1000 次迭代。表 6 显示，估计所得 Shapley值的对抗鲁棒性

表 *5.* HarsanyiNet 与可比深度神经网络每训练轮次的训练成本（秒）

数据集	HarsanyiNet	可比深度神经网络
人口普查	5.0	1.9
MNIST	243.3	127.0
CIFAR-10	205.0	106.7

表 *6.* 模型鲁棒性和 Shapley 值鲁棒性

模型	对抗样本的分类准确率（%）	ℓ_2 Shapley 值差异的范数
HarsanyiNet	13.83%	1.44
ResNet-18	8.21%	1.50

在 HarsanyiNet 上（通过 Shapley值差异的 ℓ_2 范数估计，数值越低越好）略高于 ResNet-18。

两项实验均表明，HarsanyiNet 的鲁棒性接近传统模型，甚至略高于传统模型。

F.5. Harsanyi 网络上估计的 Shapley值的更多结果

我们进行了更多实验，以展示 Harsanyi 网络所生成的解释。具体而言，我们在表格数据集上训练了 Harsanyi-MLP，在图像数据集上训练了 Harsanyi-CNN。

对于人口普查、酵母和电视新闻数据集，我们分别在图7、图8和图9中比较了各方法估算的 Shapley值。可以看出，我们的 HarsanyiNet 计算出的 Shapley值与根据定义1计算出的真实夏普利值完全相同；而各种近似方法，包括采样方法（Castro 等人，2009年）、对偶采样（Mitchell 等人，2022年）、KernelSHAP（Lundberg & Lee，2017年）以及采用配对采样的 KernelSHAP（KernelSHAP-PS）（Covert & Lee，2021年），则需要进行数千次网络推理，才能计算出相对准确的 Shapley值。

对于包括 MNIST 和 CIFAR-10 数据集在内的图像数据集，我们在图10和图11中分别生成了更多不同类别的归因图。

F.6. 计算深度神经网络编码的 Harsanyi交互强度的实验详情

当输入变量数量较少时（例如， $n < 16$），我们可以按照定义2迭代计算所有 Harsanyi交互的交互强度。对于表格数据集，包括人口普查、酵母和电视新闻数据集，我们将每个输入变量的基线值设为该变量在所有训练样本中的均值。然后，我们以穷举方式计算每个 Harsanyi交互的强度。

对于 MNIST 数据集，直接计算所有 Harsanyi交互的 Harsanyi强度并不可行。为降低计算成本，我们按照先前的工作（Ren等，2023a），从每张图像的前景中随机采样 8 个图像区域。然后，我们可以计算这 8 个采样图像区域之间所有可能的 Harsanyi交互的 Harsanyi效应（总共得到 $2^8 = 256$ 个不同的 Harsanyi交互）。至于基线值，我们将每个像素的基线值设为零。

F.7. 图4中生成归因图的实验详情

我们比较了各方法在 Harsanyi-CNN 模型上的归因图。为便于与其他方法进行比较，MNIST 数据集对应的 Harsanyi-CNN 由 4 个级联的 Harsanyi 块构成，每个 Harsanyi 块包含 $32 \times 14 \times 14$ 个神经元，其中 32 表示通道数。超参数分别设置为 $\beta = 100$ 和 $\gamma = 0.05$。CIFAR-10 数据集对应的 Harsanyi-CNN 由 10 个级联的 Harsanyi 块构成，每个 Harsanyi 块包含 $256 \times 16 \times 16$ 个神经元，其中 256 表示通道数。超参数分别设置为 $\beta = 1000$和 $\gamma = 1$。这样，HarsanyiNet 和其他四种模型无关方法使用了相同的模型，从而确保比较公平。

此外，由于 Harsanyi -卷积神经网络模型计算了特征 $\mathbf{z}^{(0)}$的 Shapley值，我们也计算了 Shapley值 on

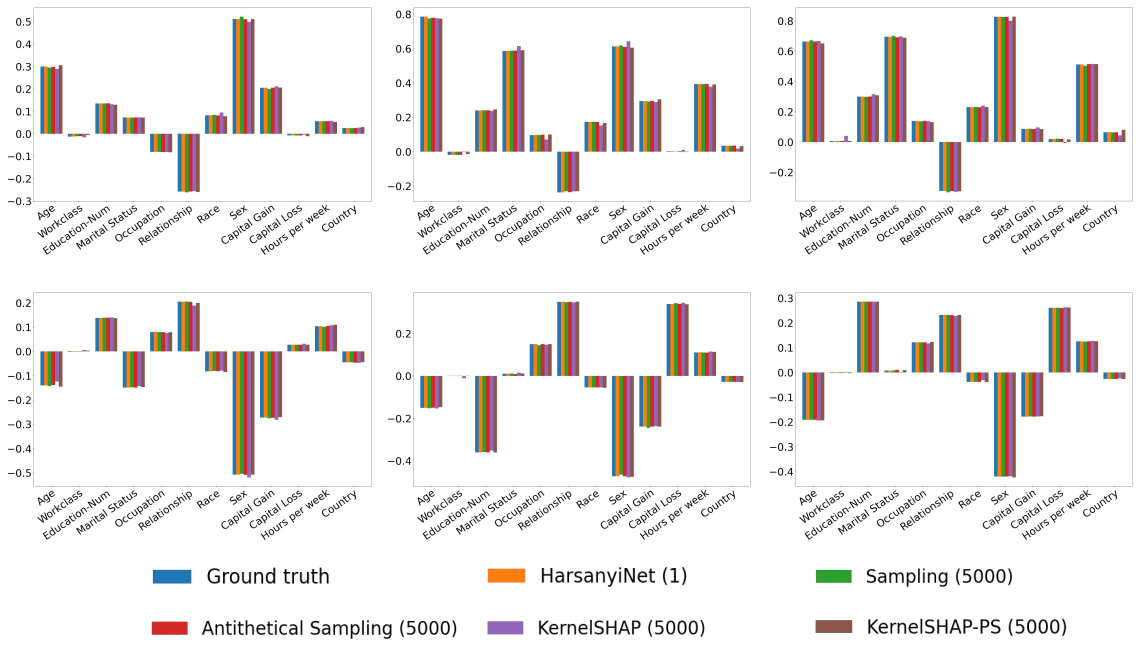


图7。在 Census 数据集上通过不同方法计算得到的 Shapley值。括号中标明了各方法执行的推理次数。第一行的样本来自类别“≤50K”，第二行的样本来自类别“>50K”。

$\mathbf{z}^{(0)}$，使用采样、KernelSHAP 和 DeepSHAP 方法。对于 MNIST 数据集，我们运行采样方法约 20000 次迭代，并运行 KernelSHAP 方法 20000 次迭代，直至收敛。对于 CIFAR-10 数据集，我们运行采样方法约 20000 次迭代，并运行 KernelSHAP 方法 200000 次迭代，直至收敛。

对于 FastSHAP 方法，我们使用训练样本$\mathbf{x}$ 和 Harsanyi-CNN 的模型预测来训练解释器模型 ϕ_{fast}，并稍微修改了模型架构，使其返回的归因图张量与$\mathbf{z}^{(0)}$的大小相同，即，对于 MNIST 数据集为 14×14，对于 CIFAR-10 数据集为 16×16。对于 MNIST 数据集，由于 Jethani 等（2021年）未报告解释器模型 ϕ_{fast}，我们训练了一个结构与 CIFAR-10 数据集所用模型相同的解释器模型，并在 MNIST 数据集上计算归因图。

F.8. 在 MNIST 和 CIFAR-10 数据集上比较计算得到的 Shapley值与真实 Shapley值的实验细节

如第4.2节所述，为了降低计算成本，我们在图像数据集上随机抽取样本前景中的 $n = 12$ 个变量作为输入变量。这样，通过掩蔽选定的 12 个变量，并将所有其他变量保留为样本中的原始变量，计算真实夏普利值。将选定变量的集合记为 $\hat{}N$，则 $|\hat{}N| = 12$。具体而言，我们将所有变量 $x_i, i \notin \hat{}N$设为基线值 $\hat{}\hat{}$，即 $\forall i \notin N, b_i = x_i$和 $\forall i \in N, b_i = 0$，从而得到基线样本 $v(\mathbf{x}_\emptyset)$。基于基线样本 $v(\mathbf{x}_\emptyset)$，我们获得了 $2^{|\hat{N}|}$ 个不同的掩码样本。对于 HarsanyiNet，根据定理1计算所选变量的 Shapley值时，我们只遍历包含所选变量的集合，即 $S \ni i, \forall i \in N$。此外，$|S|$表示 S中所选变量的数量。这样，我们通过 $\phi(i) = \sum_{S \subseteq N: S \ni i, i \in \hat{}N} \frac{1}{|S|} I(S)$ 和 $\sum_{i=1}^{n} \phi(i) = v(\mathbf{x}_N = \mathbf{x}) - v(\mathbf{x}_\emptyset)$ 计算所选变量的 Shapley值。对于 ShapNets，我们将所有变量 $\hat{}\hat{}x_i, i \notin N$设为基线值 $b_i = 0$，以获得掩码样本 $\mathbf{x}'$，即 $x'_i = \overline{x_i}, \forall i \in N$；否则为 $x'_i = 0$。然后，利用掩码输入样本，我们可以通过 ShapNet 计算所选变量的 Shapley值。

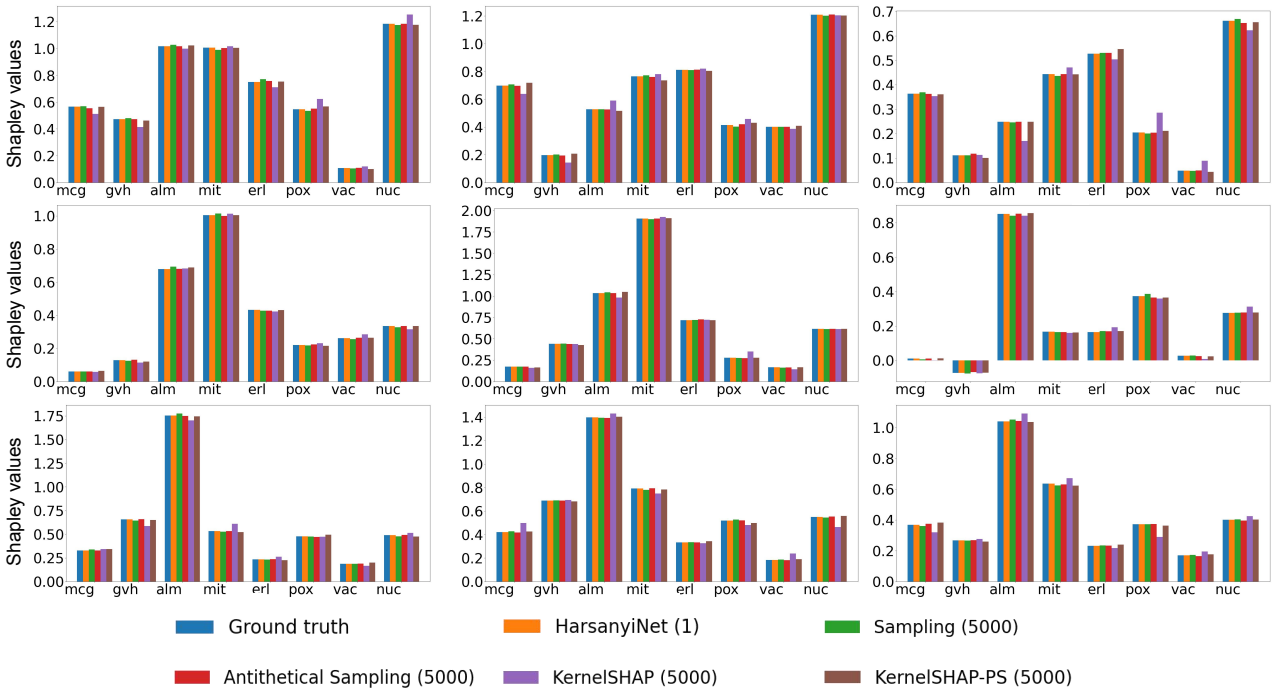


图*8.* 不同方法在酵母数据集上计算得到的 Shapley值。括号中标明了每种方法进行的推理次数。图中展示了来自 3 个类别（共 10 个类别）的样本所对应的 Shapley值。第一行样本来自类别“CYT”，第二行样本来自类别“MIT”，最后一行样本来自类别“NUC”。

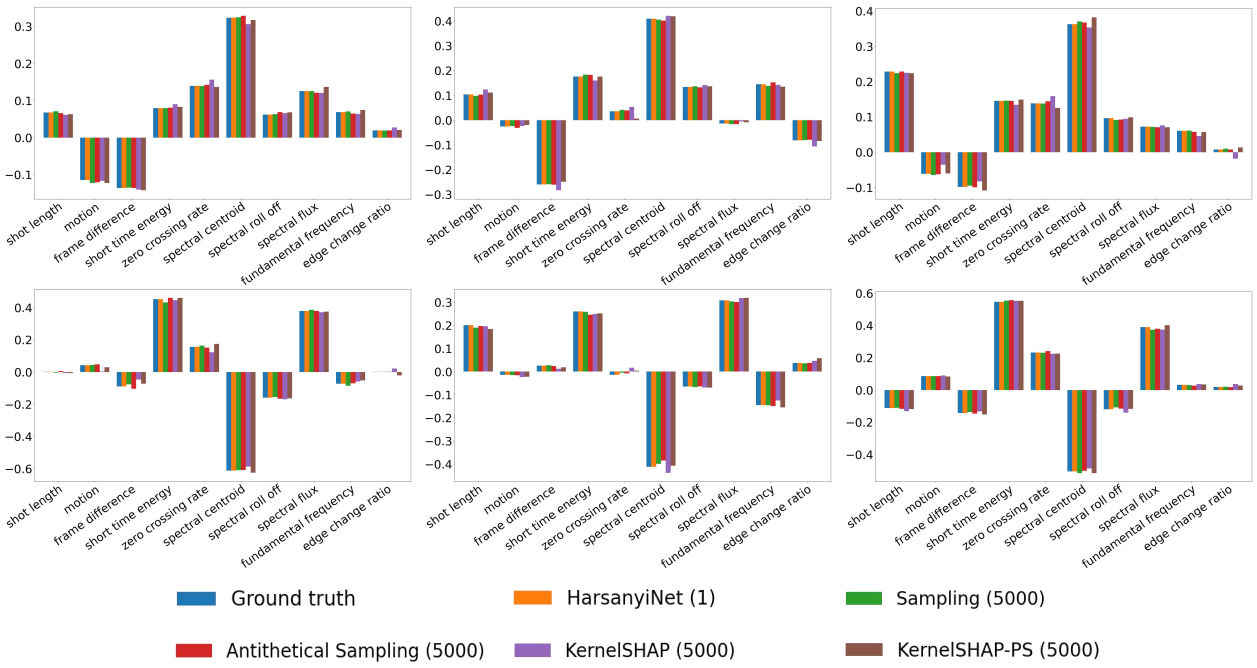


图*9.* 电视新闻数据集上不同方法计算出的 Shapley值。括号中标出了每种方法进行的推理次数。第一行的样本来自“非广告”类别，第二行的样本来自“广告”类别。

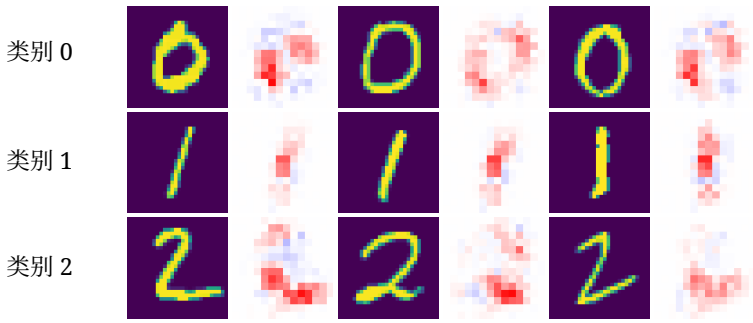


图*10.* HarsanyiNet 在 MNIST 数据集上生成的 Shapley值。将 $v(\mathbf{x}_S)$ 设置为输入样本$\mathbf{x}$的真实类别对应的输出维度，即可计算该 Shapley值。

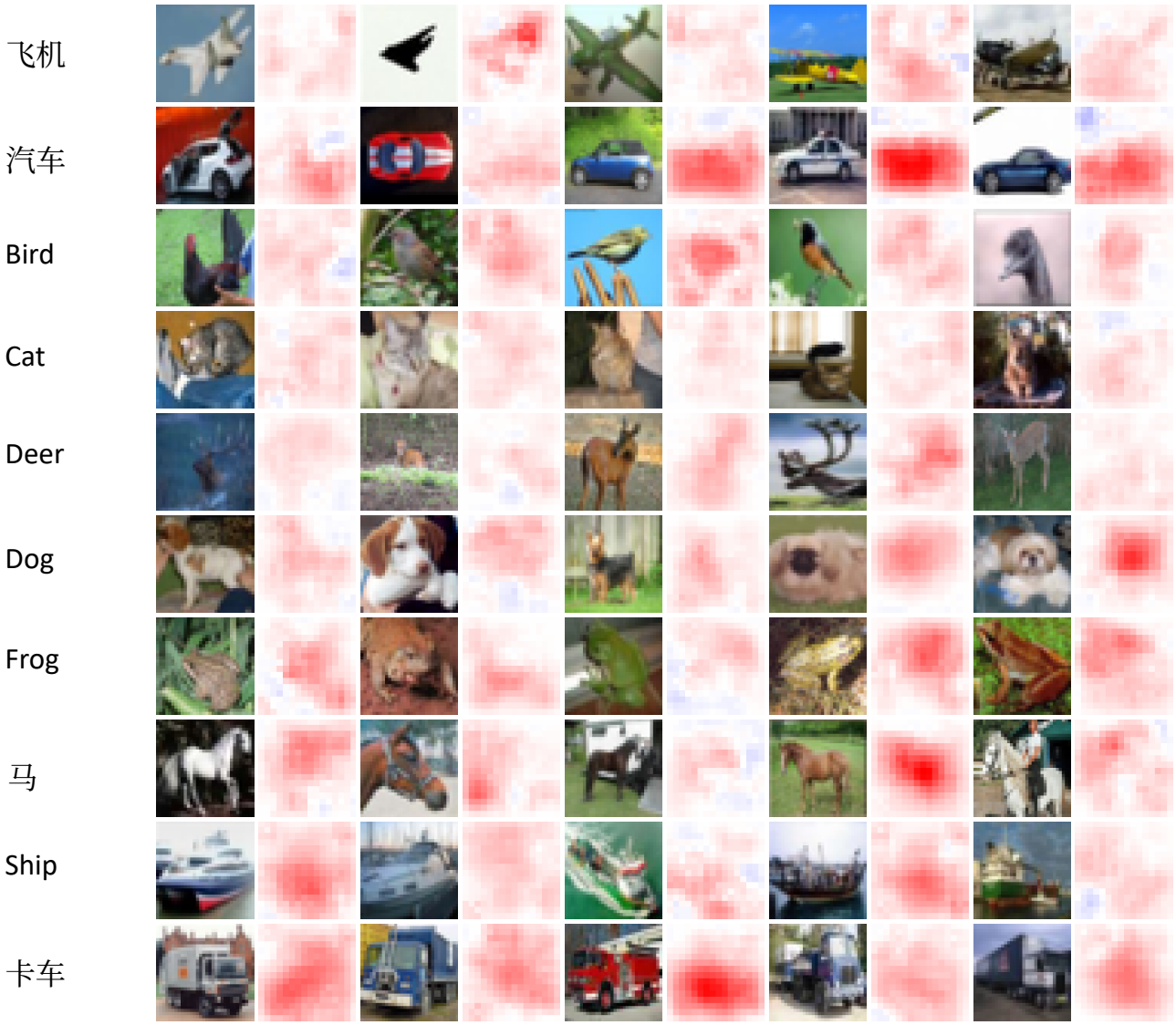


图*11.* HarsanyiNet 在 CIFAR-10 数据集上生成的 Shapley值。通过将 $v(\mathbf{x}_S)$ 设为输入样本$\mathbf{x}$的真实类别对应的输出维度，计算 Shapley值。