arXiv:2303.04935v2 [cs.CV] 5 Jun 2023

X-Pruner：面向视觉Transformer的可解释剪枝

Lu Yu 詹姆斯·库克
大学lu.yu@my.jcu.edu.au

Wei Xiang*拉
筹伯大学
w.xiang@latrobe.edu.au

摘要

近年来，视觉*Transformer*模型已成为多种任务中的主流模型。然而，这些模型通常计算成本高昂且内存需求庞大，因此难以部署于边缘平台。近期研究提出以不可解释的方式对*Transformer*进行剪枝，忽视了模型内部单元与目标类别之间的关系，从而导致性能较差。为缓解这一问题，我们提出了一种名为*X-Pruner*的新型可解释剪枝框架，该框架在设计时考虑了剪枝准则的可解释性。具体而言，为衡量每个可剪枝单元对预测各目标类别的贡献，我们提出了一种新颖的可解释性感知掩码，并以端到端方式进行学习。随后，为保留信息量最大的单元并学习逐层剪枝率，我们根据可解释性感知掩码的值，自适应地搜索逐层阈值，以区分未剪枝单元和已剪枝单元。为验证和评估我们的方法，我们将*X-Pruner*应用于具有代表性的*Transformer*模型，包括*DeiT*和*Swin Transformer*。全面的模拟结果表明，所提出的*X-Pruner*优于最先进的黑盒方法，在大幅降低计算成本的同时，仅造成轻微的性能下降。代码可在此处获取 https://github.com/vickyyu90/XPruner。

1. 引言

近年来，Transformer在自然语言处理、视觉和图等多个具有挑战性的领域中受到了越来越多的关注 [3, 9]。它由两个关键模块组成，即多头注意力（MHA）和多层感知机（MLP）。然而，与卷积神经网络类似，Transformer的主要局限在于模型规模庞大且计算成本高昂。这严重限制了它们的部署在边缘平台等资源受限设备上。为了压缩和加速Transformer模型，各种技术应运而生。常用方法包括权重量化 [30], 知识蒸馏 [28], 滤波器压缩 [24], 和模型剪枝 [21]。其中，模型剪枝，尤其是结构化剪枝，受到了广泛关注。它以硬件友好的方式移除预训练模型中最不重要的参数，因此也是本文关注的重点。

由于卷积神经网络与Transformer之间存在显著的结构差异，尽管卷积神经网络剪枝方法已取得普遍成功，Transformer剪枝研究仍处于早期阶段。现有研究大致可分为三类。(1) 基于准则的剪枝通过采用预定义准则来保留最重要的权重/注意力，例如L1/L2范数 [20], 或激活值 [6]。(2) 基于训练的剪枝利用手工设计的稀疏正则化 [31] 或资源约束 [28, 29]对模型进行重新训练。(3) 架构搜索剪枝方法基于预定义策略直接搜索最优子架构[3, 10]。尽管这些研究已取得了相当大的进展，但仍有两个基本问题尚未得到充分解决，即，最优逐层剪枝比例和权重重要性度量。

对于第一个问题，不同层剪枝率的选择会显著影响最终性能。为此，一些相关工作提出了一系列确定最优逐层剪枝率的方法 [7, 11]。例如，Michel 等人。[18] 研究了注意力头在 Transformer 自然语言处理任务中的有效性，并提出使用贪心算法剪除注意力头。Yu 等人。[28] 提出了一种剪枝算法：在保留注意力机制整体结构的同时，移除低于学习得到的逐层阈值的注意力分数。然而，这些方法没有考虑权重之间的相互依赖关系。近期，Zhu 等人。[31] 提出了 VTP 方法，通过稀疏正则化识别并移除视觉Transformer 中不重要的图像块和注意力头。然而，VTP 需要手动为所有层尝试不同的阈值。

*通讯作者。

针对第二个问题，以往研究采用各种重要性度量来识别不重要的权重，包括基于幅值的方法、基于梯度的方法 [12, 19]，和基于掩码的方法 [27]。其中，基于幅值的方法通常会导致次优结果，因为它没有考虑权重之间可能存在的相关性[24]。此外，基于梯度的方法往往会剪除值较小的权重，因为这些权重的梯度较小，反向传播可能无法将其识别为重要权重。最后，当前基于掩码的剪枝有两方面的局限性：（1）大多数基于掩码的剪枝技术会根据逐层剪枝率，针对某个单元手动分配二值掩码，效率低下且效果不理想。（2）大多数工作使用不可微掩码，导致训练过程不稳定、收敛效果不佳。

本文提出了一种面向视觉Transformer模型的新型可解释结构化剪枝框架，称为 X-Pruner。该框架通过考虑剪枝准则的可解释性，解决上述两个问题。正如可解释人工智能（XAI）领域所指出的，模型中的重要权重通常捕捉特定类别的语义信息。受这一理论启发，我们提出以按类别的方式有效量化每个权重的重要性。首先，我们为每个可剪枝单元（例如注意力头或线性层中的矩阵）设计了一个可解释性感知掩码，用于衡量该单元对预测每个类别的贡献，并且该掩码完全可微。其次，我们利用每个输入的真实标签作为先验知识来指导掩码学习，从而充分利用与每个输入相关的类别级信息。我们的直觉是，如果某个单元生成的特征表示对某个目标类别有正向贡献，那么该单元相对于此类别的掩码值就会被正向激活，反之则会被抑制。第三，我们提出了一种可微剪枝操作，并引入阈值正则化项。这使得我们能够通过基于梯度的优化搜索阈值，优于大多数采用手工设计准则剪除单元的以往研究。同时，所提出的剪枝过程可以自动完成，即，保留高于学习所得阈值的判别性单元。通过这种方式，我们以可解释的方式自动、高效地实现了逐层剪枝算法。总而言之，本文的主要贡献如下：

- 我们提出了一种新颖的可解释结构化剪枝框架，称为 X-Pruner。该框架依据可解释性，剪除对识别所有类别贡献较小的单元。据作者所知，这是首个为视觉Transformer开发可解释剪枝框架的工作；

- 我们提出为每个可剪枝单元分配一个可解释性感知掩码，以量化其贡献——于预测每个类别的作用。具体而言，所提出的掩码完全可微，并可通过端到端方式学习；

- 基于获得的可解释性感知掩码，我们提出通过可微分剪枝操作学习逐层剪枝阈值，以区分重要和不太重要的单元。因此，这一过程以可解释的方式完成；

- 我们通过全面的仿真结果表明，所提出的 X-Pruner 优于多种最先进的方法，并且在提升剪枝后模型的可解释性方面展现出优势。

2. 相关工作

2.1. Transformer剪枝

剪枝一直是移除Transformer模型中最不重要权重的一种常用方法。现有方法主要分为非结构化剪枝和结构化剪枝。对于非结构化剪枝，已有研究提出了基于幅值的方法和基于海森矩阵的方法等技术 [7, 23]。然而，这些方法会导致非规则稀疏性，产生难以与硬件效率相匹配的稀疏张量计算。

结构化剪枝可以缓解上述问题：它会移除预训练模型中信息量不足的连续结构，例如注意力头和权重矩阵的行。例如，Michel 等 [18] 发现，可以剪除很大比例的注意力头，而不会显著影响性能。Fan 等 [10]提出了一种结构化Dropout方法，在推理阶段选择模型的子结构。Wang 等 [27]使用一种参数化方法，剪除了大语言模型中的秩一分量。Liu 等 [16] 将多种模型压缩技术应用于一系列预训练语言模型，并取得了令人瞩目的成果。然而，这些工作聚焦于面向自然语言处理任务的Transformer剪枝。

对于视觉Transformer，Chen等人。 [6] 研究了非结构化和结构化稀疏性，并提出了一种一阶重要性近似方法来移除注意力头。最近，Yu 等人。 [28] 提出了一种针对视觉Transformer的结构化剪枝方法，该方法使用0/1掩码，根据模型参数的大小区分不重要和重要参数。尽管该方法使用了可微阈值，但掩码本身不可微，这可能导致梯度产生偏差，并使剩余权重的结果不理想。Yu 等人。 [29] 提出将剪枝、跳层和知识蒸馏这三种高效方法整合到一个统一框架中，以生成紧凑型Transformer。尽管

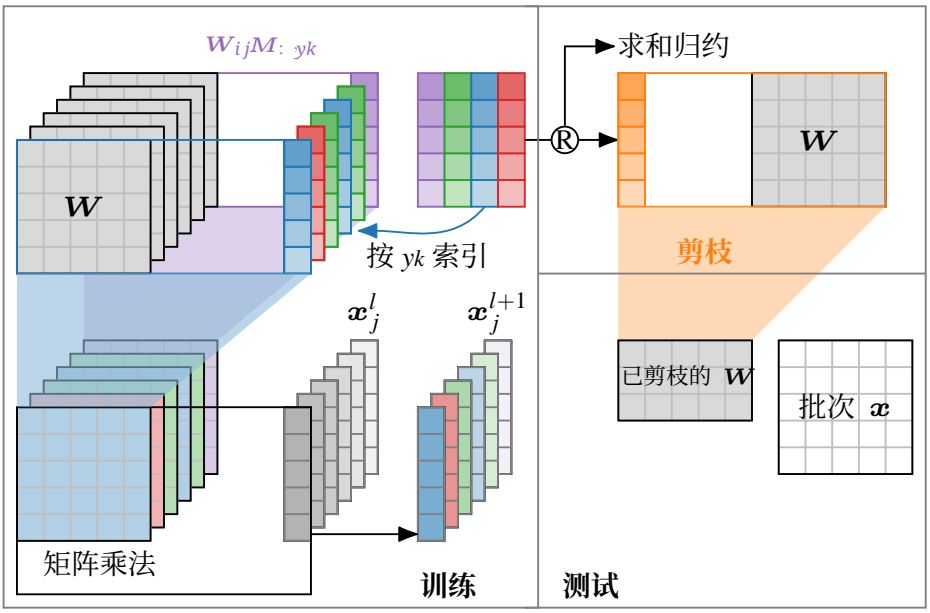

图 1. 我们提出的 X-Pruner 框架的流程。我们首先使用所提出的可解释性感知掩码训练 Transformer，旨在量化每个单元对预测各个类别的贡献。然后，我们在预定义的成本约束下探索逐层剪枝阈值。最后，对剪枝后的模型执行微调过程。

这些现有方法取得了重大进展，但重要性度量的设计仍是一个有待探索的开放问题。

2.2. 面向 Transformer 的可解释人工智能技术

在可解释人工智能方法方面，近期有许多研究致力于解释 Transformer 所作出的预测。Chefer 等人 [5] 提出了一种逐层相关性传播（LRP）方法，可同时处理正向和负向贡献。然而，这种方法无法解释自注意力之外的注意力模块。Voita等人 [26] 发现，只有少数注意力头对模型决策有显著影响。Jawahar等人 [13]证明，语义模式是在高层中获得的。Abnar 等人 [1]提出了对多个层的注意力分数进行组合的方法，但该方法无法区分正向和负向归因。Raghu等人 [22] 分析了视觉Transformer的内部表征，发现它们在所有层中学习到的表征更加统一。近期，Chefer 等人 [4] 还提出了一种通用方法，用于解释包括多模态 Transformer 在内的 Transformer。可解释人工智能技术的进步显著提升了多种任务的表现。然而，文献尚未探索可解释性在指导模型剪枝方面的能力。因此，我们提出 X-Pruner，旨在充分利用模型的可解释性，为模型剪枝推导重要性度量。

3. 方法

3.1. 问题定义

我们提出的 X-Pruner 旨在通过移除视觉 Transformer中的可剪枝单元（例如权重矩阵的行和注意力头）来探索结构化剪枝。设 $\boldsymbol{D}$为一个训练数据集，其中包含 N 个训练样本对 $\{(\boldsymbol{x}_1, \boldsymbol{y}_1).....(\boldsymbol{x}_N, \boldsymbol{y}_N)\}$。对于一个 L 层的 Transformer $f(\boldsymbol{W})$，其参数表示为 $\boldsymbol{W} = (\boldsymbol{W}^1, \boldsymbol{W}^2, ... \boldsymbol{W}^L)$，其中 $\boldsymbol{W}^l \in \mathbb{R}^{d_l}$，$1 \le l \le L, d_l$ 表示第 l 层中的可剪枝参数数量。给定目标剪枝率 α，剪枝过程可表示为剪枝率为 $\{r_l\}_{l=1}^L$ 的逐层操作：

$$(r_1, r_2, ..., r_L)^* = \text{argmin}\mathcal{L}(f(\boldsymbol{W}; r_1, r_2, ..., r_L; \boldsymbol{x}), \boldsymbol{y}),$$
$$s.t. \quad \frac{P(f(\boldsymbol{W}; \{r_l\}))}{P(f(\boldsymbol{W}))} \ge \alpha, \tag{1}$$

其中，r_l 是第 l 层的剪枝率，$P(\cdot)$ 是一种资源评估指标。

3.2. 所提出的 X-Pruner

3.2.1 可解释性感知掩码

为了充分利用类别级信息，我们为每个可剪枝单元分配一个可解释性感知掩码，用于量化每个单元对识别各个类别的贡献。具体而言，所提出的掩码是

针对每个类别的类别级掩码，而不是适用于所有类别的标量掩码。例如，给定第 l 个自注意力层中的权重由查询 $\boldsymbol{W}_l^Q \in \mathbb{R}^{n\times d}$、键$\boldsymbol{W}_l^K \in \mathbb{R}^{n\times d}$和值 $\boldsymbol{W}_l^V \in \mathbb{R}^{n\times d}$组成，其中 n 和 d分别表示输入和输出维度。头 h 的掩码表示为 $\boldsymbol{M}_{l,\,h}^H \in \mathbb{R}^{C\times d}$，其中 C 表示类别总数。也就是说，构建了$\boldsymbol{M}_{l,\,h,\,i}^H$，用于量化头 h 对识别第 i 类的贡献。显然，先前工作中使用的标量掩码是我们方法的一种特殊情况，即 $\boldsymbol{M}_{l,\,h,\,i}^H$ 的值相同。因此，给定输入 $\boldsymbol{x}_i$ 及其类别标签 $\boldsymbol{y}_i$，应用掩码时，将权重与其对应的掩码相乘。也就是说，头 h 的自注意力操作可表示如下：

$$\boldsymbol{\alpha}_{l,h} = S(\frac{(\boldsymbol{W}_{l,h}^Q \boldsymbol{x}_i)^T \boldsymbol{W}_{l,h}^K \boldsymbol{x}_i}{\sqrt{d}}), \tag{2}$$

$$\mathrm{Attn}_{l,h}(\boldsymbol{x}) = \boldsymbol{\alpha}_{l,h} \boldsymbol{W}_{l,h}^V \boldsymbol{x}_i, \tag{3}$$

$$\mathrm{MHA}(\boldsymbol{x}, \boldsymbol{M}_l^H) = \sum\nolimits_{h=1}^{H} \boldsymbol{M}_{l,h,\boldsymbol{y}_i}^H \mathrm{Attn}_{l,h}(\boldsymbol{x}), \tag{4}$$

其中，$S(\cdot)$是softmax函数，$\boldsymbol{\alpha}_h$ 是第 h个注意力权重，H 是注意力头的总数。

同时，我们将类似的思路应用于MLP和其他线性投影层。设线性层中的权重矩阵为 $\boldsymbol{W}_l \in \mathbb{R}^{m\times n}$，其中 m 和 n是其维度。相应的掩码定义为 $\boldsymbol{M}_l^F \in \mathbb{R}^{C\times m\times n}$。那么，线性层中的前馈过程可表示为：

$$\mathrm{FC}(\boldsymbol{Z}_l, \boldsymbol{M}_l^F) = \boldsymbol{M}_{l,\boldsymbol{y}_i} \boldsymbol{W}_l^F \boldsymbol{Z}_l, \tag{5}$$

其中，$\boldsymbol{Z}_l$ 是第 l层的输入。为简洁起见，我们省略所有层中的偏置。

请回想一下，我们的可解释性感知掩码旨在识别对预测标签有显著影响的权重。因此，当输入图像都属于同一类别 c时，掩码$\boldsymbol{M}_{:,\,c}$ 最好缓慢变化，从而得到平滑的可解释性感知掩码。因此，我们提出对掩码添加平滑性感知约束。具体而言，我们计算掩码值相对于输入和预测类别的二阶导数，并将其 L_1 范数作为平滑性感知约束：

$$\mathcal{L}_{\mathrm{smooth}}(\boldsymbol{M}) = \sum_{l=1}^{L}\sum_{c=1}^{C} |\nabla^2 \boldsymbol{M}_{:,c}^l|_1. \tag{6}$$

此外，为了解决可剪枝单元之间的冗余问题，我们不将所有单元都视为与模型预测相关，而是在掩码上施加以下稀疏性约束：

$$\mathcal{L}_{\mathrm{sparse}}(\boldsymbol{M}) = \sum_{l=1}^{L}\sum_{c=1}^{C} ||\boldsymbol{M}_{:,c}^l||_2. \tag{7}$$

总的来说，总损失函数定义如下：

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{sm}}\mathcal{L}_{\mathrm{smooth}} + \lambda_{\mathrm{sp}}\mathcal{L}_{\mathrm{sparse}}(\boldsymbol{M}), \tag{8}$$

其中 $\mathcal{L}_{\mathrm{ce}}$ 是交叉熵损失，λ_{sm} 和 λ_{sp} 是超参数。

与以往使用二值掩码量化各单元对所有类别贡献的工作不同，我们提出使用可微掩码来捕捉每个单元相对于各类别的重要性。训练完成后，每个已学习掩码的数值总和明确表示该单元对识别所有类别的贡献。通过这种方式，我们学习到的可解释性感知掩码以端到端方式获得了揭示 Transformer 内部推理过程的表征能力，本质上能够以直观且可解释的方式全局考察每个单元的重要性。值得注意的是，训练过程中预训练模型的权重保持不变。因此，我们通过实验证明，只需少量训练轮次即可训练所提出的可解释性感知掩码。

3.2.2 可解释剪枝

所提出的 X-Pruner 旨在剪枝后的模型中保留用于识别目标类别的最重要单元。为此，我们移除可解释性感知掩码影响最小的单元。以往的工作通过人工设定的逐层阈值来衡量各单元的重要性；由于参数搜索空间需要穷举，这种方法在计算上难以实现 [29]。在本工作中，我们设计了一种可微剪枝操作，并结合阈值正则化项来学习逐层阈值。与大多数以往研究相比，这种方法能够更好地控制非均匀稀疏性。

直观而言，借助获得的可解释性感知掩码，应剪除掩码值低于某个阈值的次要单元，同时保留重要单元。然而，目前大多数方法都使用人工设定的阈值，这很难在可训练过程中进行优化。为解决这一问题，我们提出了一种针对可解释性感知掩码的可微剪枝操作。其数学表达式如下：

$$\hat{\boldsymbol{M}}^l = \begin{cases} \boldsymbol{M}^l \tanh(n(\boldsymbol{M}^l - \theta^l)), & \boldsymbol{M}^l \in \Phi(\boldsymbol{M}^l | 1 - r^l), \\ p \tanh(n(\boldsymbol{M}^l - \theta^l)), & \text{otherwise}, \end{cases} \tag{9}$$

其中，r^l 是第 l层的剪枝率，$\Phi(\boldsymbol{M}^l|1-r^l)$ 是一个用于返回$\boldsymbol{M}^l$中排序后前（$1-r^l$）%个元素的函数。通过适当设置 n和 p，当 $\boldsymbol{M}^l \in \Phi(\boldsymbol{M}^l|1-r^l)$时，$\tanh(\cdot)$ 的值渐近趋近于 1，从而得到 $\hat{\boldsymbol{M}}^l \approx \boldsymbol{M}^l$。在这种情况下，我们提出的可微剪枝操作意味着，对识别类别贡献更大的判别性单元会被保留，

而贡献较小的单元则会被抑制。通过为 n赋予一个较大的正值，我们提出的剪枝函数能够利用反向梯度学习阈值 θ^l 。在实验中，我们通过实证验证，让 $p = 500$ 和 $n = 10$ 可确保训练过程稳定，并取得良好的剪枝效果。

随后，我们按如下方式计算所有可剪枝层的累积剪枝率：R

$$R = \sum_{l=1}^{L} \frac{r^l * n^l}{N}, \tag{10}$$

其中， n^l 表示层 l 中可剪枝参数的总数， N 表示所有未剪枝参数的数量。

为了以端到端的方式，在给定剪枝率 α 的情况下学习逐层剪枝率，我们在增广拉格朗日法中提出了一种新的正则化项 $\mathcal{L}_{\mathrm{R}}$ ，将式（1）中的优化问题转化为无约束的惩罚表达式。具体而言，它表示为

$$\mathcal{L}_{\mathrm{R}} = \beta(\alpha - R)^2 + \gamma(\alpha - R), \tag{11}$$

其中， β 和 γ 是可训练参数，式（11）中的无约束问题可以使用基于梯度下降的技术求解。总体而言，所提出的 X-Pruner 的总损失函数为

$$\mathcal{L} = \mathcal{L}_{\mathrm{ce}} + \mathcal{L}_{\mathrm{R}}. \tag{12}$$

式（12）中的优化问题使我们能够提升带有判别性掩码、对模型决策至关重要的单元，同时抑制不太重要的单元。此外，这意味着当 r^l具有更大的 n^l时，逐层剪枝率往往也更大，这对于利用所有层中的动态稀疏性而言是自然的。

训练完成后，我们根据学习得到的剪枝率 $\{r^1, r^2, ..., r^L\}$，移除影响最小的单元，并通过设置 $\boldsymbol{W} = \boldsymbol{W} * \boldsymbol{M}$，将每层剩余的、具备可解释性的掩码 $\boldsymbol{M}$整合到剪枝后的模型中，随后进一步微调剪枝后的模型。总之，我们提出的可解释剪枝方法 X-Pruner 能够以可解释且可训练的方式识别并保留重要单元，克服现有黑盒剪枝方法的缺点，并为剪枝后的模型的准确性提供实证保证。

4. 实验

为评估 X-Pruner 的性能，我们在 CIFAR-10 [14] 和 ILSVRC-12 数据集 [8]上进行实验。CIFAR-10 包含 10 个类别，由 5 万张训练图像和 1 万张验证图像组成。ILSVRC-12 包含 1K 个类别的图像，其训练集和验证集分别有 128 万张和 5 万张图像。为了与现有方法进行公平比较，我们在分类任务中对 DeiT[25] 和 Swin Transformer [17] 架构进行剪枝 [28, 29]。此外，我们还进行了一系列消融研究，以探究框架中不同组件对性能的贡献。

4.1. 实现细节

所有实验均使用 PyTorch 在 NVIDIA Tesla V100 GPU 上实现。我们使用预训练权重初始化视觉 Transformer模型，并将其作为基线模型。在训练可解释性掩码的过程中，学习率设为 0.01，批次大小为 128，并采用动量为 0.9 的 SGD 优化器。从经验来看，DeiT 的掩码训练过程为 50 个周期，Swin Transformer 为 30 个周期，耗时约 300 个 V100 GPU 小时。在可解释剪枝过程中，我们首先将所有 r^l 设为 α。θ^l 和 r^l 的学习率设为 0.02，并使用 AdamW 优化器进行微调。其他参数的学习率和动量分别为 5×10^{-4} 和 0.9。DeiT 模型训练 80 个周期，Swin Transformer 模型训练 30 个周期。除知识蒸馏外，我们遵循原始 DeiT 和 Swin Transformer [17] 所采用的训练策略。 β 和 γ 初始化为零，并在训练过程中进行优化。

4.2. 主要结果

表 1 展示了 X-Pruner 在 ILSVRC-12 上优于其他最先进方法的表现。我们观察到，大多数现有剪枝方法都无法在避免准确率大幅下降的同时，实现显著的 FLOPs 节省。相比之下，通过学习可微的可解释性感知掩码，X-Pruner 能将计算成本降低 51.3%-66.1%，而准确率仅小幅下降（0.72%-1.1%）。具体而言，在对 DeiT-T 进行剪枝时，与只能节省 46.2% 浮点运算次数的 WDPruning [28] 相比，我们提出的 X-Pruner 实现了更大的 FLOPs 节省（66.1% 对比 46.2%），且准确率下降更少（1.1% 对比 1.86%）。对于更大的模型 DeiT-S，尽管 UVC [29]在现有方法中取得了最先进的 Top-1准确率，即在浮点运算次数减少 49.6% 的情况下达到 78.82 %，但 X-Pruner 将浮点运算次数减少了 51.3%，同时 Top-1准确率达到 79.04%。这些结果表明，所提出的 X-Pruner 优于现有剪枝方法，能够获得更紧凑的模型尺寸和更好的性能。

同时，我们研究了所提出的方法在另一种热门 Transformer 即Swin Transformer [17]上的有效性。实验结果见表2。对于 Swin-T，X-Pruner 带来了显著

表 1. 在 ILSVRC-12 数据集上与最先进方法的比较。浮点运算次数保留比例表示浮点运算次数与完整模型浮点运算次数之比。 $*$ 表示在训练过程中使用了知识蒸馏。

模型	方法	Top-1 准确率（%）	Top-5 准确率（%）	浮点运算次数（G）	浮点运算次数保留比例（%）
DeiT-T	基线	72.2	91.10	1.3	100
	（NeurIPS'20）SCOP [24]	68.9	89.00	0.8	61.5
	（ICCV'21）HVT [21]	69.7	89.40	0.7	53.8
	（ICLR'22）UVC* [29]	70.6	-	0.5	39.1
	(AAAI'22) WDPruning [28]	70.3	89.82	0.7	53.8
	X-Pruner	**71.1**	**90.11**	**0.6**	**49.2**
DeiT-S	基线	79.8	95.00	4.6	100
	（NeurIPS'20）SCOP [24]	77.5	93.50	2.6	56.5
	（ICCV'21）HVT [21]	78.0	93.83	2.4	52.2
	（ICLR'22）UVC* [29]	78.82	-	2.3	50.4
	(AAAI'22) WDPruning [28]	78.38	94.05	2.6	56.5
	X-Pruner	**78.93**	**94.24**	**2.4**	**52.1**
DeiT-B	基线	81.8	95.59	17.6	100
	（NeurIPS'20）SCOP [24]	79.7	94.50	10.2	58.3
	（ICLR'22）UVC* [29]	80.57	-	8.0	45.5
	（AAAI'22）WDPruning [28]	80.76	95.36	9.9	56.3
	X-Pruner	**81.02**	**95.38**	**8.5**	**48.5**

表 2. Swin Transformer 在 ILSVRC-12 数据集上的剪枝结果。

	方法	Top-1	浮点运算次数	Top-1 ↓	浮点运算次数 ↓(%)
Swin-T	基线	81.2	4.5	0.0	0.0
	STEP [15]	77.2	3.5	4.0	22.2
	ViT-Slim [3]	80.7	3.4	0.5	24.4
	X-Pruner（我们的方法）	**80.7**	**3.2**	**0.5**	**28.9**
Swin-S	基线	83.2	8.7	0.0	0.0
	STEP [15]	79.6	6.3	3.6	27.6
	WDPruning [28]	81.8	6.3	1.4	27.6
	X-Pruner（我们的方法）	**82.0**	**6.0**	**1.2**	**31.0**

更高的 Top-1 准确率，同时浮点运算次数大幅减少。具体而言，我们的方法减少了 28.9% 的浮点运算次数，而 Top-1 准确率仅下降 0.5%。在对 Swin-S 进行剪枝时，与考虑剪枝维度的最先进方法 WDPruning [28] 相比，得益于可解释性感知掩码的使用，我们的 X-Pruner 也展现出显著优势。

4.3. 可视化与分析

我们使用基于LRP的相关性方法，展示基于DeiT-S及其剪枝后模型生成的类别级视觉解释图 [1]。图2基于随机选取的 ILSVRC-12 验证图像进行了可视化比较。从图中可以看出，完整模型的大多数视觉解释结果在人看来仍像是噪声状图案。然而，WDPruning [28] 和 UVC [29]得到的剪枝后模型所生成的图均发生了失真。尽管剪枝后的模型预测正确，但剪枝过程之后生成的解释图却不正确。相比之下，我们发现，X-Pruner 剪枝后模型生成的视觉解释图更加紧凑，噪声也更少。

此外，图中所示的注意力层学习得到的掩码值 3 表明，所提出的 X-Pruner 无需逐层剪枝率即可恰当地发现各注意力头的重要性。值得注意的是，高层（第 11 层和第 12 层）的掩码值高于第 2 层的掩码值。这表明，在 Transformer 中，低层同时关注局部和全局信息，而高层则关注全局信息。因此，高层捕获了丰富的语义级特征，这些特征对于最终预测至关重要。

我们进一步将 X-Pruner 与最先进的方法 WDPruning 进行比较 [28]，比较对象为 CIFAR-10。图 4 展示了 DeiT-S 在不同剪枝率下的 Top-1准确率。如图所示，在较低的剪枝率下（例如 10%），两种方法的准确率都略高于基线。当剪枝率较高时，与 WDPruning [28], 相比，在相同剪枝率下（例如 50% 或 70%），我们的 X-Pruner 准确率损失更小。

4.4. 消融研究

在本小节中，我们首先基于 Swin-T 模型评估所提出方法中可解释性感知掩码的有效性

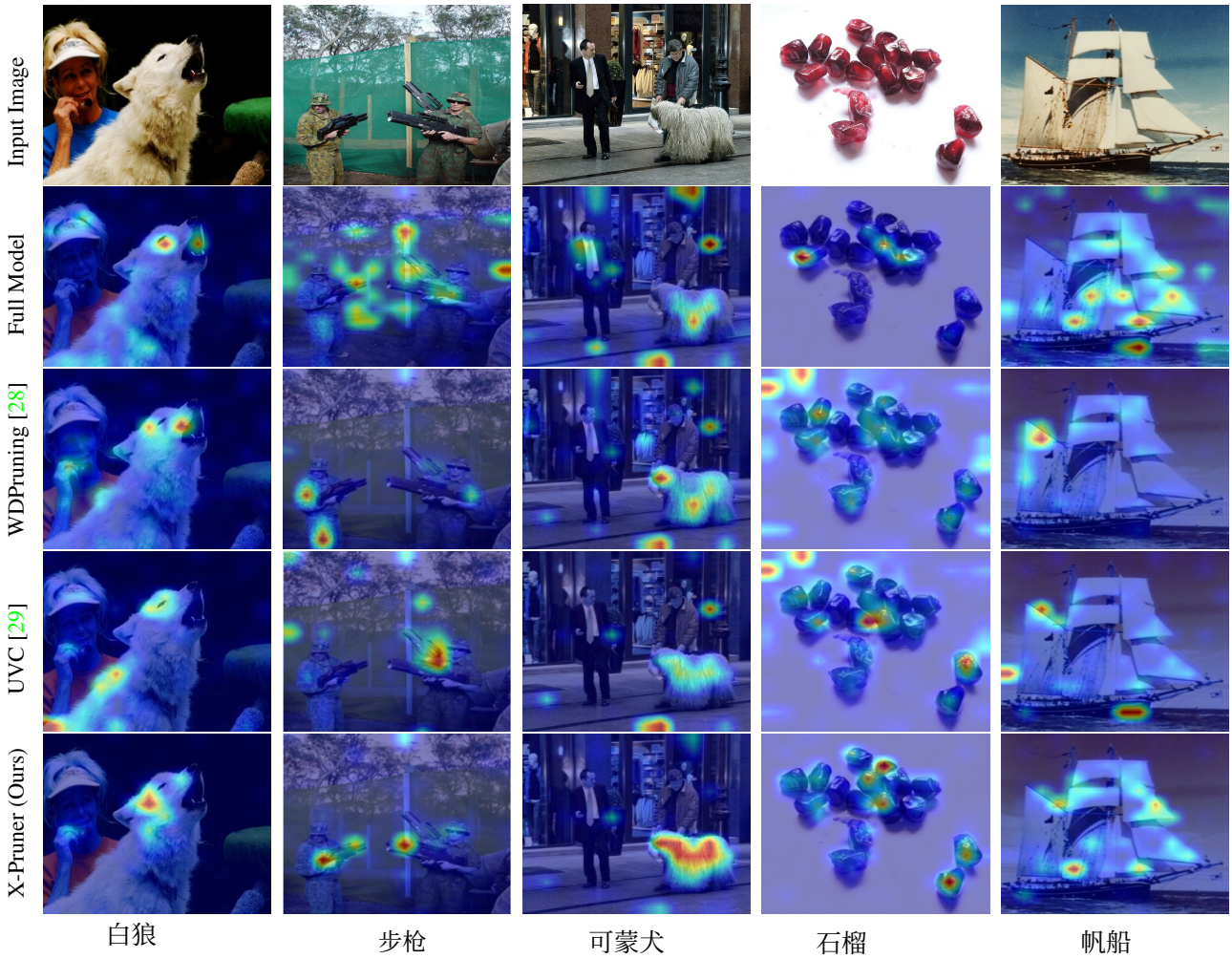

图2. 在 ILSVRC-12 验证集上，由各种剪枝网络生成的视觉解释。从上到下依次为：输入图像、原始 DeiT-S 的视觉解释图、由 WDPruning 剪枝的模型 [28], UVC [29], 以及我们的 X-Pruner 生成的视觉解释图。

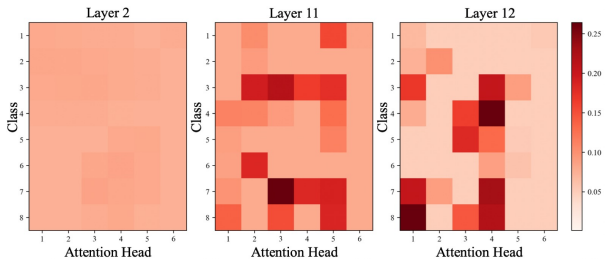

图3. DeiT-S 各层的可解释性感知掩码值。

。表 3 展示了详细结果。为保证公平比较，所有模型均采用相近的浮点运算次数剪枝率进行剪枝。我们首先采用类别无关策略训练可解释性感知掩码，并将其记为不使用可解释性感知掩码。也就是说，对于不同类别的所有输入，都使用同一个掩码。然而，由于

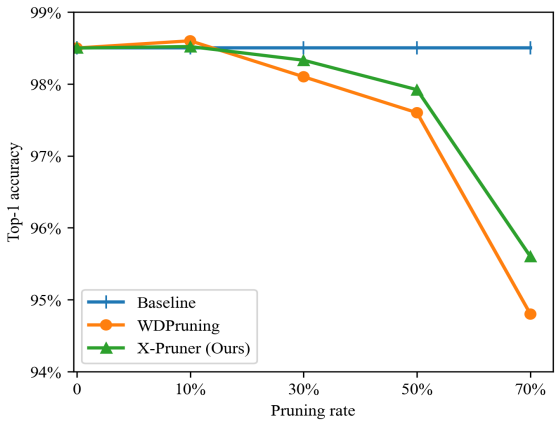

图4. DeiT-S 在 CIFAR-10 上不同剪枝率下的 Top-1准确率。“基线”表示未经剪枝的基线模型。

设置	Top-1 ↓(%)	浮点运算次数 ↓ (%)
无掩码	2.65	28.9
无 $\mathcal{L}_{\text{平滑}}$	1.02	29.3
不含 $\mathcal{L}_{\text{稀疏}}$	1.92	29.1
X-Pruner	**0.51**	**28.9**

表 3. 在 ILSVRC-12 上采用不同配置对 Swin-T 进行剪枝的主要结果。

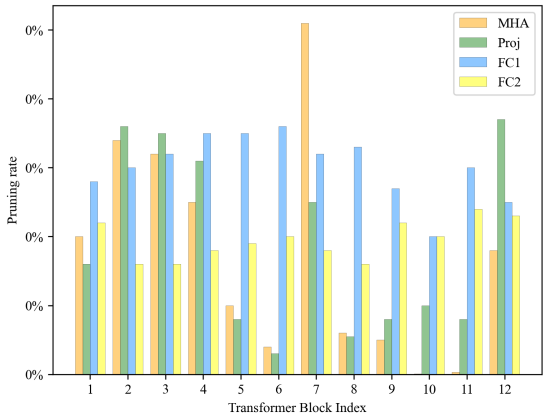

图 5. 当 DeiT-S 的剪枝率设为 0.3 时，各模块中单元的剪枝率。

方法	Top-1 ↓ (%)	浮点运算次数 ↓ (%)
随机剪枝	2.28	47.2
均匀剪枝	4.05	47.4
X-Pruner	**0.87**	**47.9**

表 4. DeiT-S 上可学习剪枝率的主要结果。

它失去了用于识别每个单元贡献的类别级信号。我们进一步探究了优化约束的影响。此外，如表3所示，当训练掩码时不使用稀疏正则化项 λ_{sparse}，训练后的模型 Top-1 准确率下降了 1.92%。这证明我们的方法有效缓解了过拟合问题，并提升了性能。最后，若移除平滑约束 λ_{smooth}，Top-1 准确率会下降 1.02%。总的来说，我们提出的 X-Pruner 方法能够有效剪枝模型，同时保持令人满意的准确率。

在表4中，我们进一步研究了 ILSVRC-12 上的逐层剪枝率，并将其与随机剪枝和均匀剪枝进行比较。在我们的方法中，每一层要剪除的单元数量会根据全局预算自适应地确定。均匀剪枝的 Top-1 准确率下降了 4.05%。我们还将随机剪枝应用于 DeiT-S，但其性能同样较差。最后，我们提出的 X-Pruner 以较小的 Top-1 准确率降幅（0.87%）优于这两种方法。

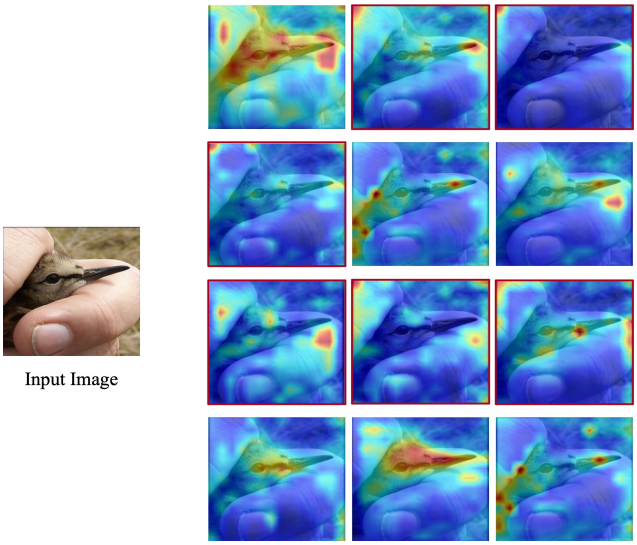

图 6. DeiT-B 第 4 层生成的注意力图可视化。红框表示根据我们学习到的掩码值剪掉的注意力头。

在 Top-1 准确率仅下降 0.87% 的情况下，表现优于这两种方法。

我们在图5中展示了 DeiT-S 的逐层剪枝率。我们发现，我们的方法能够结合可解释性感知掩码值，自动学习剪枝后的架构，这优于估计各个可剪枝单元的重要性。此外，通过在图6中可视化 DeiT-B 模型第 4 层生成的注意力图，我们发现，所提出的 X-Pruner 确实移除了主要关注背景、对最终预测贡献较小的冗余注意力头。

5. 结论

本文提出了 X-Pruner，这是一种新颖的可解释 Transformer 剪枝框架。在 X-Pruner 中，我们提出了一种新颖的可解释性感知掩码，用于评估每个可剪枝单元对预测每个类别的贡献。该掩码完全可微，并通过我们提出的类别级正则化项进行学习，以缓解过拟合。随后，我们引入了一种新的可解释剪枝过程，逐层学习剪枝率，直至达到资源约束。大量实验表明，从模型可解释性的角度来看，X-Pruner 能够显著降低多种 Transformer 的计算成本。此外，X-Pruner 仅以轻微的准确率下降为代价，便超越了最先进的剪枝方法。

参考文献

[1] Samira Abnar 和 Willem H. Zuidema。量化 Transformer 中的注意力流。在 *ACL*，第 4190–4197 页，2020年。

3, 6[2] Alejandro Barredo Arrieta、Natalia D´ıaz Rodr´ıguez、Javier Del Ser、Adrien Bennetot、Siham Tabik 等。可解释人工智能（XAI）：概念、分类体系、机遇与挑战，以及迈向负责任的人工智能。在《信息融合》，58:82–115，2020年。2[3] Arnav Chavan、Zhiqiang Shen、Zhuang Liu、Zechun Liu、Kwang-Ting Cheng 和 Eric P. Xing。视觉Transformer瘦身：在连续优化空间中进行多维搜索。在 CVPR，第 1 11, 2022 1, 6 – 页。[4] Hila Chefer、Shir Gur 和 Lior Wolf。用于解释双模态和编码器—解码器 Transformer 的通用注意力模型可解释性。在 ICCV，第 387 396, 2021 3 – 页。[5] Hila Chefer、Shir Gur 和 Lior Wolf。超越注意力可视化的 Transformer 可解释性。在 CVPR，第 782–791 页，2021年。3[6] Tianlong Chen、Yu Cheng、Zhe Gan、Lu Yuan、Lei Zhang 和 Zhangyang Wang。追寻视觉 Transformer 中的稀疏性：端到端探索。在 NeurIPS，第 19974–19988 页，2021年。1, 2[7] Tianlong Chen、Jonathan Frankle、Shiyu Chang、Sijia Liu、Yang Zhang、Zhangyang Wang 和 Michael Carbin。预训练BERT网络的彩票假设。在 NeurIPS，第 1–13 页，2020年。1, 2[8] Jia Deng、Wei Dong、Richard Socher、Li-Jia Li、Kai Li 和 Li Fei-Fei。ImageNet：大型分层图像数据库。在 CVPR，第 248–255 页，2009年。5[9] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、Xiaohua Zhai、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly 等。一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer。在 ICLR，第 1 19 页，2021年 1 – 。[10] Angela Fan、Edouard Grave 和 Armand Joulin。通过结构化Dropout按需缩减 Transformer 深度。在 ICLR，第 1 15, 2020 1, 2 – 页。Elias Frantar 和 Dan [11] Alistarh。SPDY：具有加速保证的精准剪枝。在 ICML，第 6726 6743, 2022 1 – 页。[12] Lun Huang、Wenmin Wang、Jie Chen 和 Xiaoyong Wei。用于图像描述的注意力之上的注意力。在 ICCV，第 4633–4642 页，2019年。2[13] Ganesh Jawahar、Benoˆıt Sagot 和 Djam´e Seddah。BERT 学到了哪些语言结构？在 ACL，第 3651 3657, 2019 3 – 页。A. Krizhevsky 和 G. [14] Hinton。从微型图像中学习多层特征。技术报告，多伦多大学，2009年。5[15] Jiaoda Li、Ryan Cotterell 和 Mrinmaya Sachan。Transformer 注意力头的可微子集剪枝。《计算语言学协会汇刊》，第 1442–1459 页，2021年。6[16] Yuanxin Liu、Zheng Lin 和 Fengcheng Yuan。ROSITA：采用集成技术的改进型BERT压缩。在 AAAI，第 8715–8722 页，2021年。2[17] Ze Liu、Yutong Lin、Yue Cao、Han Hu、Yixuan Wei、Zheng Zhang、Stephen Lin 和 Baining Guo。Swin Transformer：采用移位窗口的分层视觉Transformer。在 ICCV，第 9992–10002 页，2021年。5

[18] Paul Michel、Omer Levy 和 Graham Neubig。十六个注意力头真的比一个好吗？发表于 NeurIPS，页码 14014–14024，2019年。1, 2[19] Byunggook Na、Jisoo Mok、Hyeokjun Choe 和 Sungroh Yoon。通过代理数据加速神经架构搜索。发表于 IJCAI，页码 2848–2854，2021年。2[20]Bowen Pan、Rameswar Panda、Yifan Jiang、Zhangyang Wang、Rog´erio Feris 和 Aude Oliva。IA-RED^2：面向视觉 Transformer 的可解释性感知冗余消减。发表于 NeurIPS，页码 24898–24911，2021年。1[21] Zizheng Pan、Bohan Zhuang、Jing Liu、Haoyu He 和 Jian- fei Cai。采用层次池化的可扩展视觉 Transformer。发表于 ICCV，页码 367–376，2021年。1, 6 [22] Maithra Raghu、Thomas Unterthiner、Simon Kornblith、Chiyuan Zhang 和 Alexey Dosovitskiy。视觉 Transformer 的视觉方式是否与卷积神经网络相同？发表于 NeurIPS，页码 12116–12128，2021年。3[23] Victor Sanh、Thomas Wolf 和 Alexander M. Rush。移动剪枝：通过微调实现自适应稀疏性。发表于 NeurIPS，页码 1–12，2020年。2[24] Yehui Tang、Yunhe Wang、Yixing Xu、Dacheng Tao、Chun- jing Xu、Chao Xu 和 Chang Xu。SCOP：用于可靠神经网络剪枝的科学控制。发表于 NeurIPS，页码 1 11，2020 1, 2, 6 – 。[25] Hugo Touvron、Matthieu Cord、Matthijs Douze、Francisco Massa、Alexandre Sablayrolles 和 Herv´e J´egou。通过注意力实现训练数据高效型图像 Transformer 及蒸馏。发表于 ICML，页码 10347–10357，2021年。5[26] Elena Voita、David Talbot、Fedor Moiseev、Rico Sennrich 和 Ivan Titov。分析多头自注意力：专门化的注意力头承担主要工作，其余的可以剪枝。发表于 ACL，页码 5797–5808，2019年。3[27] Ziheng Wang、Jeremy Wohlwend 和 Tao Lei。大语言模型的结构化剪枝。发表于 EMNLP，页码 6151–6162，2020年。2[28] Fang Yu、Kun Huang、Meng Wang、Yuan Cheng、Wei Chu 和 Li Cui。视觉 Transformer 的宽度与深度剪枝。发表于 AAAI，页码 3143–3151，2022年。1, 2, 5, 6, 7[29] Shixing Yu、Tianlong Chen、Jiayi Shen、Huan Yuan、Jianchao Tan、Sen Yang、Ji Liu 和 Zhangyang Wang。统一视觉 Transformer 压缩。发表于 ICLR，页码 1–17，2022年。1, 2, 4, 5, 6, 7[30] Qingru Zhang、Simiao Zuo、Chen Liang、Alexander Bukharin、Pengcheng He、Weizhu Chen 和 Tuo Zhao。PLATON：基于权重重要性上置信界剪枝大型 Transformer 模型。发表于 ICML，页码 26809–26823，2022年。1[31] Mingjian Zhu、Yehui Tang 和 Kai Han。视觉 Transformer 剪枝。发表于 KDD，页码 1–4，2021年。1