Efficient Transformer

蔡文朴
南京大学计算机科学与技术系

Efficient Transformer

- 优化模型设计：
- 优化方法设计：
- 优化目标设计：
- 优化变量设计：

- 优化模型设计

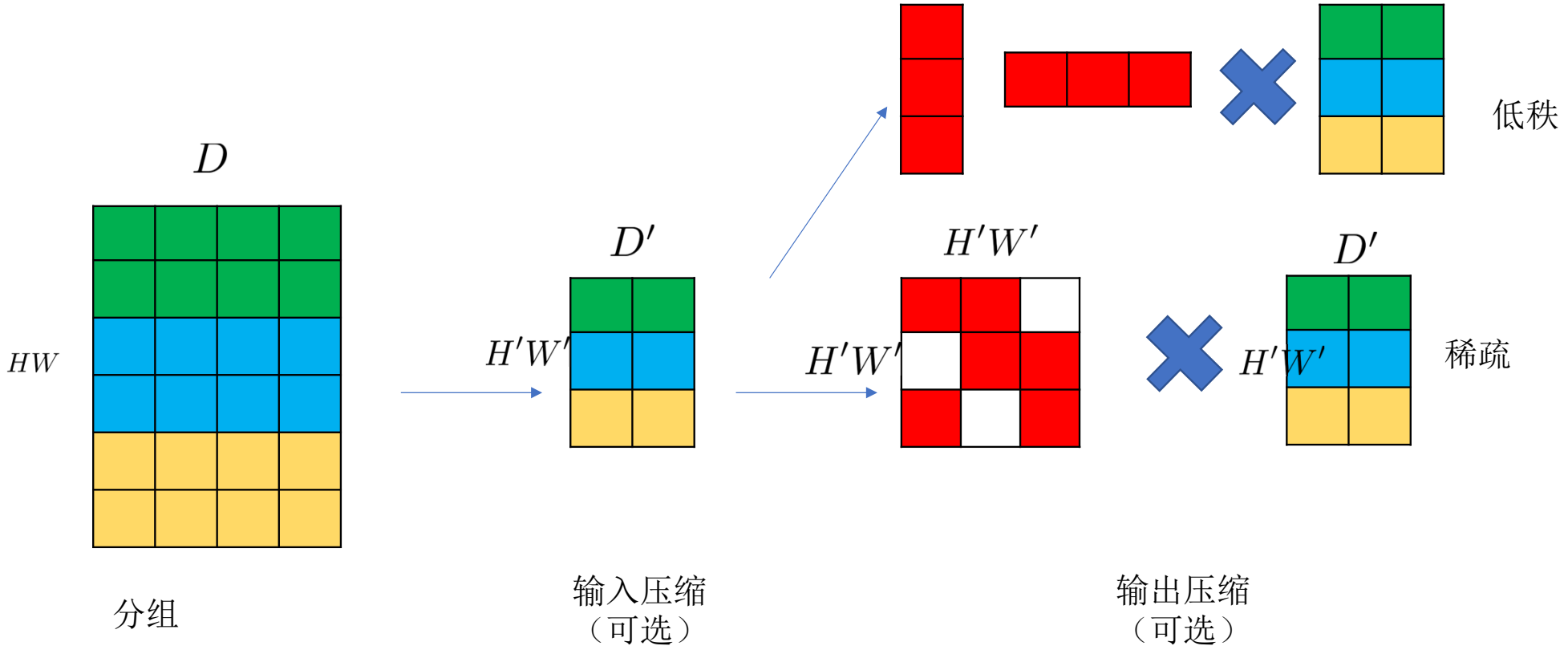
优化模型设计

- 1 总览：权衡拟合能力和高效
  - 每层特征分为多组
  - 组内采用相同的计算模块，每组只针对某一种表达能力建模
  - 组间采用交互算法，多种表达能力融合
  - 在计算量给定的情况下，建模更多表达能力
- 2 压缩模式
- 3 组合压缩
- 4 应用任务

优化模型设计-总览

- 1 总览：权衡**拟合能力和高效**
  - 权衡**高效+拟合能力（权衡）**
    - 显示计算的部分->隐式的近似计算：SGH Performer
    - 并行、异构、串行、指导、结合压缩：不同模块不同的压缩方法，每个模块在尽可能高的压缩率下，尽可能得到较高的**表达能力**
      - 稀疏+低秩注意力：Scatterbrian LS-Transformer
      - 稀疏+量化模型参数：DJPQ
      - Sparse-Max 和 Softmax 组合形成 ent-max
  - 权衡**高效+泛化能力（权衡）**
    - 鲁棒性+泛化：DQ-ICLR'2020
  - 增强模型对GPU的友好度（**软硬协同**）
  - 无需训练的压缩和微调的压缩结果的不一致性
    - 微调的学习率太小，陷入了局部最优

优化模型设计-总览



2 压缩模式

压缩可以去除一些冗余/无关/干扰信息，提升模型性能

- 2-1 维度，维度内部又分为不同的粒度
  - 2-1-1 序列/空间位置：序列剪枝 分辨率压缩（**空间冗余**）
    - **水平维度，垂直维度**
  - 2-1-2 时间位置：帧剪枝 （**时间冗余**）
  - 2-1-3 网络宽度，特征/通道：（**特征冗余**）
    - 粒度：头剪枝 特征降维 通道剪枝 注意力矩阵压缩
    - 尾部类别剪枝-ADATTL
  - 2-1-4 网络深度：层剪枝 块剪枝-多层剪枝 （**语义冗余**）
  - 2-1-5 参数维度：增强重构能力（参数冗余）
    - 粒度：连接，滤波器
  - 2-1-6 多输入：多分组，多模态，多实例，多样本，多视图，多分支（**输入冗余**）
  - 2-1-7 数据集维度：每次迭代选择部分数据用于训练
- 2-2 压缩粒度-分组：沿着某一维度分组，以组为单位压缩；采用选择矩阵进行分组
  - 2-2-1 分组方法（Assign获取）
  - 2-2-2 分组度量
  - 2-2-3 分组模式
  - 多分组对应多种分组大小、方法、度量、模式，但都是沿着同一维度
- 2-3 压缩向量
  - 2-3-1 输入压缩
  - 2-3-2 输出压缩
  - 2-3-3 内积压缩

2 压缩模式

- 2-4 压缩方法
  - 选择矩阵
    - 硬注意力or软注意力
    - 稀疏矩阵压缩：不规则稀疏 分块稀疏 N:M稀疏（**稀疏重构**） **硬注意力**
    - 低秩矩阵压缩：权重共享、量化、矩阵分解（**低秩重构**）**软注意力**
  - 重构矩阵
- 2-5 静/动态压缩
  - 动态压缩 ：不同的样本走不同的压缩策略，节省计算量，更好平衡精度和速度
    - 动态+计算量分配：
      - 每个样本对于每个分支的计算量不同，可以为0，也可以为最大，也可以为中间：
    - 动态压缩维度
      - 样本自适应配置各个维度的压缩率
      - 动态组合：MIAformer
    - **动态压缩方法**（增强权衡能力）
      - 样本自适应通道/序列剪枝：FBS
      - 样本自适用层剪枝：SkipNet AIG
    - **训练无法加速，batch版本的推理无法加速，**
  - 静态压缩：省去动态计算模块，节省计算量
    - 固定参数
    - 固定参数+选择矩阵
- 2-6 数据相关or数据无关
  - 数据相关：参数化上采样deconv；

2 压缩模式

- 2-7 计算量分配：给定总计算量每个分支/分组分配多少计算量（权衡）
  - 每层的特征维度：
    - 固定：BERT VIT Reformer FBS
    - 层次：逐渐增加维度 ResNet
    - 交替：瓶颈结构BottleNeck MobileBERT，反瓶颈结构MobileNetv2 FFN，先增后减：Delight
    - 压缩率->可学习参数：Slimming
  - 每层的序列数目：
    - 固定：VIT Dvit Evit
    - 层次：逐渐递减 PVT Pit
    - 压缩率->可学习参数：
      - 搜索每个分支/分组的剪枝率：PowerBERT
      - 静态推理 搜索每一层的gnum，适应不同的数据集
    - 问题：第一层不剪枝会造成推理内存瓶颈

2 压缩模式

- 2-8 额外参数共享方式，引入额外参数提取任务/标签/分支特定特征
  - 序列门控
  - 模块门控
  - 参数门控
  - 索引门控
  - 组合门控
- 2-9 多种指标，压缩的模型要从多个方面满足多个指标
  - 低存储
    - 降低存储开销，序列压缩不会减少参数
  - 高速模型/低延迟
    - 训练：
      - 训练后压缩
      - 精度/loss预测器：节省训练的eval的时间；对于离散的输入(e.g.剪枝率)，非离散的输入也能给出一个近似结果OFA QFA-NIPS'21 CC-CVPR'21
    - 推理
      - 全整数量化
      - 累加器量化
      - 避免乘法操作
  - 低内存
    - 推理：避免太多连接边
    - 训练：固定部分分支，采样部分分支
  - 给定精度损失的压缩
    - 保证模型的误差控制在给定范围内

2-1 压缩维度

- 2-1 维度，维度内部又分为不同的粒度
  - 2-1-1 序列/空间位置：序列剪枝 分辨率压缩（空间冗余）
    - 水平维度，垂直维度
  - 2-1-2 时间位置：帧剪枝 （时间冗余）
  - 2-1-3 网络宽度，特征/通道：（特征冗余）
    - 粒度：头剪枝 特征降维 通道剪枝 注意力矩阵压缩
    - 尾部类别剪枝-ADATTL
  - 2-1-4 网络深度：层剪枝 块剪枝-多层剪枝 （语义冗余）
  - 2-1-5 参数维度：增强重构能力（参数冗余）
    - 粒度：连接，滤波器
  - 2-1-6 多输入：多分组，多模态，多实例，多样本，多视图，多分支（输入冗余）
  - 2-1-7 数据集维度：每次迭代选择部分数据用于训练

2-1 压缩维度

- **2-1-2 特征维度：特征低秩/通道剪枝，head聚合：**
  - 选择矩阵，单个or多个：
    - Head聚合：one-head-NIPS'2019 ATTR-AAAI'21 EBERT Fishformer-NIPS'22
    - 降维：
      - DSA WaveVit-ECCV '22
      - 池化：CBAM MobileVIT-v2
      - 维度=类别数：PC-Net-CVPR'20
    - 稀疏：只选择部分维度 CG-NET SPL FBS-ICLR'19 DGC Manidp-CVPR'21
- **2-1-3 参数维度：稀疏/非结构化剪枝**
  - 结构化的参数压缩对应到序列/特征压缩
  - 参数分组：CUP Cluser-Reg SCSP
  - 静态： Tetris Block-Sparse-BERT Tile-sparse GKP
  - 动态：
    - 动态权重：
      - Carafer-ICCV'19 HyperSeg-CVPR'21
      - Hash-MOE-NIPS'21 DY-Net-CVPR'20 DCD-ICLR'21 DDF-CVPR'21

2-1 压缩维度

- **2-1-4** 层聚合/层压缩，串行分支聚合
  - 单分支：
    - 动态：AIG SKIPnet Deebert PABEE CAT
    - 多层共享 AL-BERT Mini-VIT
  - 多分支：Msdnet RA-Net DVT GF-Net （分类层不适用细粒度的层，只能用于粗粒度的层）
  - 多层输出聚合：
    - 多层特征聚合：FPN PVT Swin HS-Net
    - 多层注意力矩阵聚合：
      - EA-ICML'21 DCA-Net-ECCV'20 GET-AAAI'21 BA-Net-ECCV'22 CABVIT-2022
      - 用于解码器：PPFormer-MICCAI'22
- 2-1-5 帧压缩
  - TSN Mgsampler
- 数据集维度：
  - Tri-level：选择预测结果越来越好的样本

2-1-6 多输入聚合

- 多个实例、样本、切块、视图、模态: 多输入融合方法（**多输入**）
- 重点在融合方法、模块设计
  - 多切块特征：医疗图像 cityscape
  - 多视图特征：
    - 多摄像头，同一个病人多个片子
    - BEV
  - 多模态特征：
    - **融合方式**
      - **序列聚合：**
        - VLT-ICCV'21 MTF-VIT-CVPR'22 CEN CAM Multimodalformer VATT-NIPS'21 CMX'22
      - 特征融合：
        - 注意力：Film, Two-Stream-Video MBT
      - 模态选择：
        - AdaMML-ICCV'21
      - 特征对齐：
        - RGB+optical：HFAN-ECCV'22 BATMAN-ECCV'22

2-1-6 多输入聚合

- 多模态特征
  - 模态设置
    - 文本+图像：
      - 整体图像：SSAH GI-Net-ECCV’20 CLIP Biomedclip-2023
      - 区域图像：ClipSeg-CVPR’22 GroupVit-CVPR’22 DenseCLIP-CVPR’22 GLIP-CVPR’22 GroundDINO-23
      - 区域+整体：Xdecoder-CVPR’23
    - 标签ID+图像：语义特征（标签）+ 原始特征融合：语义特征只能是数据集为单位的
      - 拼接标签token和特征特征，作为Q/K：Segdeformer-ECCV’22 Segmentor
    - 图像+图像
      - MTF-CVPR’22 Multi-MAE-ECCV’22
    - 通用模态：ImageBind-CVPR’23

2-1-6 多输入聚合

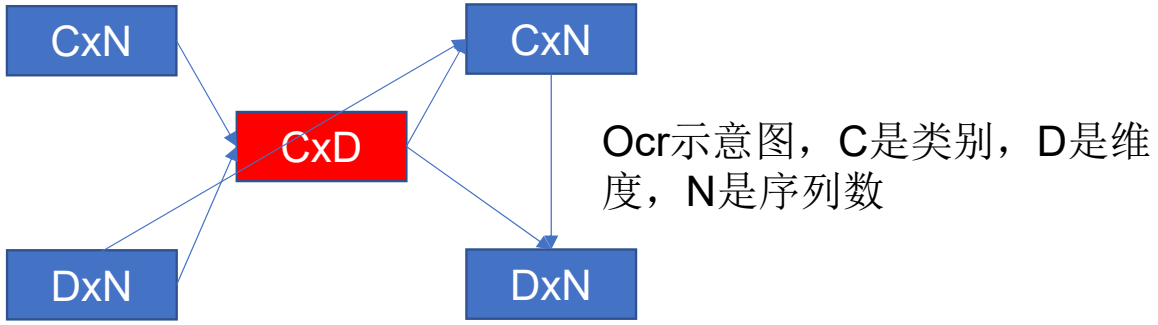
- 多分支聚合：多个分支加权（可以去除部分分支）合成一个
  - 都可以分为两步
    - 拼接/相加/相乘：
      - 相乘：Faster-RCNN PraNet URIM UAM RF-Net MASK2former
      - 拼接：OCR UACA BAM
    - 自注意力（交叉注意力看作拼接+自注意力，其中的一部分），保留需要精炼的分支
  - 静态：
    - 直接拼接：DVT
  - 动态：判别性融合，选取判别性特征/分支
    - **输入是原始的多尺度特征，输出是融合后的多尺度特征/一个总特征**
    - 串行融合：
      - 编码解码：FPN DDSC-CVPR'18 TransUNET Swin-Unet Uformer（**多尺度+最大尺度高层语义，小尺度缺乏高层语义**）
      - 多层聚合：D3Net-cvpr'21
      - 层次融合：一次性融合距离较远的编码、解码特征，语义差距太大
        - 层次精炼底层特征：Unet++
      - **+级联串行**：PA-Net-CVPR'18 EfficientDet-CVPR'20
    - **并行融合（注意力）：**
      - 直接相加/拼接：
        - SparseMLP-AAAI'22 HireMLP ViP MorphFC Container SegNext （**多尺度+多感受野**）PA-Net
        - +多层参数共享：MRN-MM'2020
        - 加权融合，避免深层网络过平滑：Cait Deepvit
    - 串行+并行：Poolnet-CVPR'19
    - +多维度聚合
      - Transfuse：特征+空间

2-1-6 多输入聚合

- 判别性融合：
- 作用一：语义对齐，缓解高层底层特征语义信息不一致，包括多层级encoder特征和encoder-decoder特征的语义对齐
- 作用二：保留部分预训练特征，防止过拟合下游任务
  - 强调更重要的分支：分支维度(交叉)注意力，每个位置or样本选择不同的分支(基本都是动态的)
    - MOE：TLC-CVPR'2022
    - 多个ROI分支，动态参数：Sparse-RCNN
    - 多个BN分支，每个分支负责一个压缩率：Slimmable
    - 硬注意力(分支选择) Revit Nommer Structure-LTH
    - 全部分支参与：Unet3+-ICASSP'20 Scaleformer-IJCAI'22 UCTrans-AAAI'22 FSFomre-ACCV'22 GFF-AAAI'22
      - Unet和FPN每次融合只使用了encoder的其中一支，这一支是语义gap较大的一支
    - 类别/置信度特定分支：每个类别的feat看作一个分支
      - OCR ACF UACA BAM Segmentor
  - 强调更重要的特征：特征维度(交叉)注意力
    - 多分支交叉注意力
      - SK-Net ResNest SparseFinder Reformer SMYRF Lite-HR-Net Focal IPE DWN Bvit Senformer
      - DFNNet-CVPR'19 UCTrans-AAAI'22
    - 分支自注意力：
      - Epsilon-resnet-CVPR'18 Mixformer-CVPR'22 Conformer-ICCV'21 Mobileformer Adafilter-AAAI'20
  - 强调更重要的位置
    - 多分支交叉注意力：Attn-UNET FPT-ECCV'20 PAN Nommer FaPN LiteHR ACSNet
      - 额外的位置mask分支(粗类别mask/粗置信度mask)：Mask2fomer URIM Pranet
    - 分支自注意力：DetectoRS-cvpr'21 Abs-ViT-CVPR'23

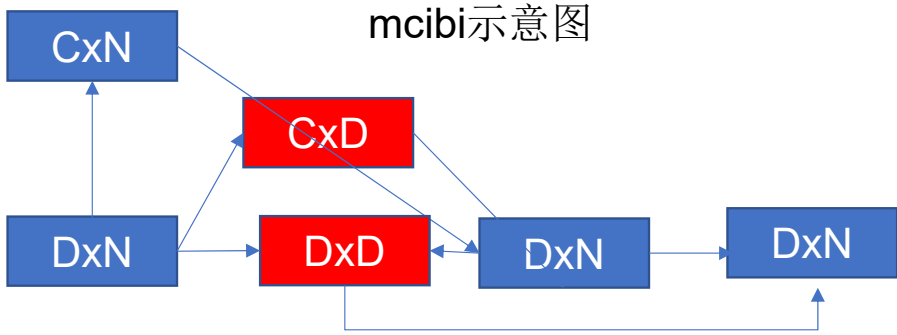
2-1-6 多输入聚合

- 判别性融合：
  - 精炼分支选择：精炼所有分支or精炼部分分支
    - 所有分支：SK-Net OCR (第一次attn精炼类别分支，第二次attn精炼feat分支)
    - 单一分支： Attn-UNET Urim Maskformer
    - 注：FCN的分类head，类别特定分支就是权重，这时没有对类别特定分支精炼
  - 多次融合串行：
    - 单次：OCR UACA
    - 多次：Mask2former urim
  - 基于模型的融合：HSNet-ICCV'21 VAT-ECCV'22



Ocr示意图，C是类别，D是维度，N是序列数

mcibi示意图



2-1-6 多输入聚合

- 多分支聚合：多个分支加权（可以去除部分分支）合成一个；可以是分支输出的分类结果，这时可以用于建模不确定性
- 增大分支数目：
  - FPN在最高层可以再产生多尺度特征，EDN-TIP22
- 动态：
  - 局部融合：空间位置一一对应，对角注意力
    - 特征维度不一致or空间位置不是一一对应，对齐多种特征在一对一融合：（对齐）
      - 上采样特征和底层特征空间位置对齐：最近邻上采样导致高分辨率和低分辨特征在边界位置不一致
        - SF-Net-ECCV'20 FaPN IFA-ECCV'22 SCRNet-AAAI'21
        - DynamicHead-nips'2020
      - 特征维度对齐：FitNet
      - Coat VIL
  - 全局融合：空间位置不一一对应，全局注意力
    - 多个不同位置的不同尺度特征融合，（多位置融合）
      - 比如一个物体的多个部分-头和身体，一个物体的上下文-鼠标键盘辅助屏幕识别 Dynamic-Head-nips'20
      - FPT-ECCV'20

2-1-8 组合维度聚合

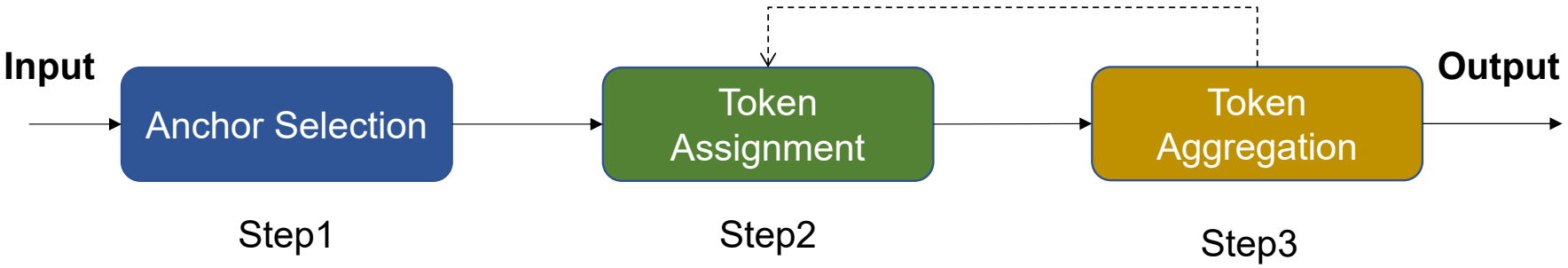
- **序列+特征组合维度压缩**
  - 序列低秩+特征低秩：ScalableVIT

2-2 压缩粒度-分组

- 2-2 压缩粒度-分组：沿着某一维度分组，以组为单位压缩；采用选择矩阵进行分组
  - 2-2-1 分组方法（Assign获取）
  - 2-2-2 分组度量
  - 2-2-3 分组模式
  - 多分组对应多种分组大小、方法、度量、模式，但都是沿着同一维度

2-2 压缩粒度-分组

- 分组维度（压缩粒度+压缩维度）：
  - 按维度分为多组（多个Head）
  - 按序列分组
  - 按层分组（层剪枝）
  - 模型分组（多模型剪枝）
  - 多个输入分组：W和A， Q和K和V，多个样本（Tracking），编码输入和解码输入
- 一个输入->多个划分；一个划分 -> 多个组；一个组 -> 多个序列or多个特征
- 分组方法：
  - Anchor初始化
  - 度量函数
  - Assign获取
  - Token聚合
- 问题：对于非N^2的注意力，特征距离的计算无法忽视
  - 度量函数：降维
  - Assign获取：只计算和部分anchor的距离；部分token不计算assign，提前去掉


Input
Anchor Selection
Token Assignment
Token Aggregation
Output
Step1
Step2
Step3

2-2 压缩粒度-分组

- 分组方法（对应Assign获取）： 采用分组矩阵（重排矩阵）分组，每个分组结果构成一个划分，也就是**Token**连接图的获取
  - 类中心查找一个/几个样本：不会出现空桶，且可以保持桶平衡：
    - 不可重复分组：
      - 降维+量化：SMYRF
      - 量化：Reformer Mongoose
      - TP Cluster-Former 贪心算法找最近的类簇进行排序，相当于寻找TSP问题的近似解
    - 可重复分组：
      - PermuteQuant Routing Boat-2022 EDPT-NIPS'22 SIT-ECCV'22 Svit'22
      - 局部重叠等分：3x3-s1 pooling Channel-Net Swin T2T CVT SPT
  - 样本查找最近的一个/几个类中心，可能出现空桶：
    - 一对一：K-means TOME-ICLR'23-setB作为类中心 TPS-CVPR'23
    - 一对多：全查找不会出现空桶：GMM EDPT
  - Assign放松到[0,1]：谱聚类
    - hard_assign放缩到soft_assign：MinCutPool NDP
    - 问题：还是需要assign初始化
  - 贪心：适用于多类簇，不会出现空桶，且可以保持桶平衡（第一步anchor选择就省略了）
    - 自顶向下，只适用类簇少的：Bi-kmeans BOAT：层次聚类，每层平衡分为两组
    - 自底向上，适用类簇多的：Agglom DPC+AC SCC-KDD'21 SEP-ICML'22
    - 自顶向下+自底向上： MultiLevel
    - 问题：贪心方式不容易平行，速度慢，无法应用到模型加速

2-2 压缩粒度-分组

- 分组度量函数
  - 特征距离：（全局空间上下文-全局向量相似冗余）
    - 一维特征距离：
      - 桶号距离： Reformer Mongoose Cluster-Former ACT
      - 数轴距离： SMYRF
    - 高维特征距离：
      - 汉明距离：Cluster-Attn
      - 欧式距离：K-means K-mediod Tcformer Kmedoid-VIT
    - 相似度向量-聚类：谱聚类 HCTransformer
    - 核函数空间距离：极限情况是1NN，也就可以发现任意形状的类簇
  - 重要性得分：Evit，得分相近的聚到一起，因此小得分都聚集到了一起
  - 空间距离：（局部区域上下文-局部空间相似冗余）
    - 2d空间的位置距离：
      - DGE KAT-MICCAI'22
      - Hard-coded CGNL GE-Net Sinkhorn
      - 2D局部分组： VIT PIT HBO-Net Token-Pooling Pvt, Rest, Segformer Pvtv2 ESVit Nommer Swin Pooling-Former DGE PCAA-cvpr'22
      - 3d局部分组：Vivit UNETR VideoSwin
    - 水平垂直距离：
      - Sparse-Former Axial-deeplab CC-Net AFM
    - Strided距离：GG
    - 全局排序：Tetris NM-Permute Sinkhorn

2-2 压缩粒度-分组

- 分组度量函数
  - 降维方法：加速距离计算，降维使得样本距离更有区分性，缓解维度灾难
    - 静态：全局的降维向量+量化函数：
      - Mongoose Cluster-Former Ecoformer-NIPS'22
    - 动态：样本特有的降维向量+量化函数
      - Reformer SMYRF Cluster-Attn ACT YOSO-ICML'22
  - 多种分组度量：
    - 重要性分组+ 聚类分组: FCABert-ACL'22 PnpDERT
    - 先验分组 + 先验分组：DGE
    - 先验分组+聚类分组
      - Mine ： 结果变差？
      - SLIC Kcenter-Video-ECCV'22

2-2 压缩粒度-分组

- 分组模式
  - 动态分组：
    - 动态分组方法：每个样本的分组方法是否相同 （增强+表达能力）
      - 参数分组（静态）：CUP Cluser-Reg SCSP
      - 特征分组（动态）：SMYRF
  - 分组大小：(这里每个分组内token数目默认一致)
    - 可学习的分组大小： DPT Adaptive-Span VSA
    - 手动设计的分块大小：EIT，HiLo
    - 动态分组大小：每个样本的分组大小不一致
      - 序列聚类
      - 可学习的分组大小：DPT-MM'21
    - 多种分组大小混合
      - UVC：head粒度+通道粒度
      - JCW-ECCV'22：通道粒度+连接粒度
    - 分组大小限制：Coke-ICML'22
  - 分组数目
    - 预先给定：一般来说分组越少，组内计算是一致的，计算越粗糙，结果越 差
      - 减少分组，增大矩阵大小，增大并行度：Hydra-2022 RTFormer-NIPS'22
    - 动态桶数目：DGConv ACT
    - 指定类簇数目or Not
      - 指定：K-means DPC
      - 不指定：Mean-shift DBSCAN
  - 参数化分组模块or非参数化分组模块

2-2 压缩粒度-分组

- **压缩粒度+压缩维度：**
  - token(H,W维度)的分组：多个token分到一组，聚合多个token
    - Attn是一层所有token一组
    - Token聚类 Clusterr-attn SMYRF
    - 多组token一个bit：DTQattn
  - Channel维度的分组：多个维度分到一组，聚合多个维度
    - FFN是一层所有维度一组
    - Head切分，维度按顺序分组，最后用**拼接**聚合
    - 卷积，一层所有维度一组，卷积用**加法和拼接**聚合多组结果（所以卷积输出维度可变）
      - 产生K^2 * N_out组HW输出，每K^2组移位加法聚合，得到N_out组结果
      - 分组卷积，多个维度一组，组的大小可以不一样，每组的输出通道也可以不一样
        - MobileNet CondenseNet GKP
      - 深度卷积，每个维度一组，深度卷积用**拼接**聚合多组结果
    - 分组剪枝：SSL CUP
    - 计算维度和维度的注意力：Bilinear-Pool Xcit
    - N:M稀疏+分组：Permute-NM Permute-OVW-NIPS'22
  - 层维度分组：跨层分组
    - 串行层：
      - 连续两层的所有token分到一组:EA-ICML'2021
      - 剪枝：OICSR Syn_Strength-NIPS'18
    - 并行层：
      - 剪枝：CURL GBN ASSL-NIPS'21（mask对齐） SRPN
  - **多维度(H,W,C)混合稀疏分组：**包含Shift
    - 空间+时间：TSM
    - 空间+特征：Cycle-MLP AS-MLP S2-MLP SPACH MS-MLP

2-2 压缩粒度-分组

- 嵌套分组：分组矩阵 + 再分组矩阵
  - 多维度：特征分组+序列分组
    - Reformer SMYRF Routing：Head分组，head内对序列分组
    - Cswin VIP PALE: 水平+竖直+Head特征
- 平衡分组or不平衡分组：各个分组的数目是否相同
  - 不平衡：
    - K-medoid-ViT VSA-ECCV'22
  - 平衡：类中心找样本
    - SeLA-ICLR'20
- 连通性保障分组：
  - SLIC，保证同一类簇的token在空间上是相连的
- 多种划分+多维度压缩+嵌套分组

2-2 压缩粒度-分组

- 分组矩阵（重排矩阵）-聚类
  - 优化目标：
    - 基于中心的：最小化类内样本-中心聚类,
      - **Ward**
      - **Mean-shift**
    - 基于最小割：最小化类内样本-样本距离，最大化类间样本-样本距离 （二者是互补的）发现任意形状的类簇
      - 谱聚类
      - Average-link
    - 最小类内最近邻距离，发现任意形状的类簇
      - Single-link
      - DBSCAN 密度聚类
    - 最小类内最远邻距离 Complete-link

2-3 压缩向量

- 2-3 压缩向量
  - 2-3-1 输入压缩
  - 2-3-2 输出压缩
  - 2-3-3 内积压缩

2-3-1 输入压缩

- 输入稀疏模式
  - 不规则稀疏
    - KVT-ECCV'22 topk-Attn
  - 分块稀疏
    - 行列稀疏：AutoCompress-AAAI'20 BlockPruning KGS StructADMM YOLOBile-AAAI'21
  - N:M稀疏
    - 1xN-Priuning

2-3-1 输入压缩-不规则稀疏

- https://github.com/oresths/tSparse
  - 非结构化稀疏
- https://github.com/rusty1s/pytorch_sparse/blob/master/csrc/cuda/spmm_cuda.cu
  - 非结构化稀疏，在矩阵较小时，50%的稀疏率也能加速，速度快；但是矩阵较大时dense计算速度更快；且大矩阵的dense计算速度比小矩阵的dense计算快
  - 提供了Pytorch调用Nvdia kernel的接口
- 哈希
  - https://github.com/mlpen/YOSO
  - Fast-Transformer：速度很慢
- QuadTree
  - 速度很慢
- **3d点云**
  - **MinkowskiEngine**
  - GitHub - traveller59/spconv: Spatial Sparse Convolution Library
- traveller59/spconv
  - 非结构化稀疏 int8量化
- Facebook/SparseConvNet: 非结构化稀疏

2-3-1 输入压缩-分块稀疏

- https://github.com/openai/triton:（xformers, Scatterbrain MLPruning使用 ）
  - 块稀疏只能是固定的模式
- https://github.com/huggingface/pytorch_block_sparse：
  - 速度慢，80%的稀疏才达到dense的水平
  - Nn_pruning的加速来自行和列的剪枝
  - 误差较大
- https://developer.nvidia.com/blog/accelerating-matrix-multiplication-with-block-sparse-format-and-nvidia-tensor-cores/
  - 声称50%的稀疏率达到dense水平，C++难以调用
  - blockELL 不支持Batch版本的计算
  - 只支持同构模式
- Dfss attn：**DFSSATTEN: Dynamic Fine-grained Structured Sparse Attention Mechanism**
  - 只支持128的块大小，只支持同构的模式，序列512时速度提升不明显(10%提升)
  - 2:4 稀疏待测试
- https://github.com/ceruleangu/Block-Sparse-Benchmark/blob/master/cusparse/test.cu
  - cusparseSbsrmm 不支持batch版本的
- TVM
  - https://github.com/lmbxmu/1xN/blob/master/model_tune.py
  - 待测试 **tvm.relay.nn.sparse_dense**
- 多种分块检索
  - **https://github.com/VITA-Group/Structure-LTH**
- 行列压缩，异构压缩
  - https://github.com/clevercool/TileSparsity
- FlashAttn：待测试

2-3-1 输入压缩- N:M 稀疏

- https://developer.nvidia.com/blog/exploiting-ampere-structured-sparsity-with-cusparselt/ :
  - 2:4结构化稀疏，在A100上，声称1.5倍速度对比dense水平，实际测试没有任何加速
  - C++难以调用
- XNN-PACK：https://github.com/google/XNNPACK
  - 没找到稀疏代码，只有量化代码
- Deformable卷积
- 平衡稀疏：EIE DARB
- 量化+稀疏：https://github.com/gilshm/sparq

2-3-1 输入压缩 其他库

- Power-iter 求特征向量：
  - https://github.com/cvlab-epfl/Power-Iteration-SVD/
- Faiss 检索，哈希，聚类；支持GPU；不支持Batch版本；Pytorch兼容性差；支持平衡聚类？
- RepLKNet：加速大kernel的卷积

2-3-1 输入压缩-选择矩阵

- 选择矩阵：得到原型，这里都是整体为单位压缩
- 硬注意力or软注意力，(M ,N)的注意力矩阵
  - 硬注意力，注意力取{0,1}（token信息损失）重要信息/部分信息增强，避免无关信息干扰
    - 静态：
      - 固定位置剪枝 PSVIT-CVPR'22
    - 空间位置剪枝：LC SB-NET DYN-Conv Patchdrop Anytime
    - 数据库剪枝： DPFP
    - **量化共享**：也可以看做硬注意力
      - 多头共享KV: LambdaNet
      - 多组共享QK： MSA
      - 低精度：Sanger DMI FG

2-3-1 输入压缩 硬注意力or软注意力

- **硬注意力，注意力取{0,1}**（token信息损失）重要信息/部分信息增强，避免无关信息干扰
  - 序列剪枝：选择重要性得分高的token
    - 非参数化：
      - 随机样本作为类中心：Kmediod Kmeans
      - 等分位置的样本：
        - SMYRF Reformer
      - 熵作为重要性得分：
        - 本身特征：STT Informer-AAAI'21 QS-ATTN VTC-LFC-NIPS'22
        - 组内特征组合：TOKEE
      - 最小重构误差排序：
        - 输入 PowerBert LTP ATS KVT-ECCV'22 Evovit Evit TopK-attn VIG-NIPS'22; Sparse-VIT-CVPR'23
        - 参数 CAP FPGM Cluster-Attn ACT PyramidBERT;
      - 循环一致性得分：q_i的最近邻是k_j，k_j的最近邻也是q_i
        - CycleGAN CyCTR
      - 2d空间均匀anchor：
        - 无需插值，就是token本身：Vit HVT PVT，NOME 50%token作为acnhor，因此一次压缩最多降低50%的token
        - 近邻插值得到：SILC EDPT; 不能整除的HVT
      - 局部近邻：
        - DPC DPC-KNN ADPC TCformer
      - 最远距离（覆盖整个特征空间）PointNet++ Kmeans++

2-3-1 输入压缩 硬注意力or软注意力

- 硬注意力，类中心**anchor初始化**：可以在token中选择，也可以不在token中选择
  - +参数化：
    - 最小Loss，可学习位置偏移：RepGraph-ECCV'20 Deformable-DETR-ICLR'21 PSVIT-ICCV'21 DAT-CVPR'22
    - 最小Loss，可学习得分，需要Gate+STE：
      - token独立计算得分：SP-VIT-eccv'22 Ada-vit-cvpr'22 SVITE-NIPS'21 IA-RED，DualAttn
      - 注意力聚合+得分计算：Dvit SAG-Pool ASAP
    - 辅助loss得到粗分类特征，无法端到端训练
      - Faster-RCNN Deformable-DETR DDQ-CVPR'23 FastInst
    - 参数化类中心，Proxy： Routing NCM-ACL'22 ProtoSeg-CVPR'22
  - 迭代初始化：
    - 迭代多次学习，逐渐精炼：PS-VIT-ICCV'21
    - 贪心，每次选择部分anchor：
      - 迭代的最短距离，每次只有部分token的得分是确定的： PyramidBert K-means++ CenterClip
    - 部分替换：PAM
    - 迭代优化：LLR-SDS-CVPR'16
  - 组合方法：
    - 重要性得分 + 距离当前集合 ： Tcformer

2-3-1 输入压缩 硬注意力or软注意力

- **软注意力，注意力取[0,1]**（多个token语义混淆，弱化重要token）
  - 静态：速度快，引入了先验信息
    - 频域转化
      - 傅里叶变换 FFT：增强全局信息
        - GF-NET AFNO FEDFormer-ICML'22 Ynet-MICCAI'22
      - 波变换：
    - 固定参数：
      - Asym-Nonlocal-ICCV'19 APCNet Glore-CVPR'19 Linformer Funnel Patch-Slim
    - 分辨率压缩：
      - RANet-CVPR'20 IC-NET
  - 双随机矩阵，行归一化+列归一化
    - 序列压缩：Sinkhorn
    - 维度压缩：CBP-CVPR'16 MoNet-CVPR'18: 特征哈希用于维度压缩

2-3-1 输入压缩 硬注意力or软注意力

- 软注意力，注意力取[0,1]， Token聚合（多个token语义混淆，弱化重要token）
  - 原型聚类（聚类得分获取）：速度慢，捕获非局部的信息
    - 重要性得分：
      - 最小loss，可学习得分：LIP Tcformer SSBNet-ECCV'22
      - 重构误差：
        - 注意力得分：EVIT Evo-vit
        - 欧式距离：CenterFormer Token-Learner EMA-ICCV'19 Constell-ICLR'21 PMM-ECCV'20 EDPT-NIPS'22 TP SVIT'22
      - 已有得分：Softpool
    - 特征降维：降低的维度作为压缩后的序列数
      - A2Net-NIPS'18 PnpDetr SIT-ECCV'22 DynaMixer-ICML'22 Paca-CVPR'23
      - 注意力聚合+降维： Diffpool-NIPS'18
    - 均匀近邻：
      - TP(K-means K-medoid) ACT DGT Cluster-Attn LISA
    - 平衡注意力：行的L0范数相等，属于每个原型的样本数相等
  - 矩阵分解（后图输入由X变为字典）
    - 双线性池化参数：FBN-ICCV'17 Attn-Pool-NIPS'17
    - 模型通用参数：ALBERT LANR Tucker-BERT DictFormer LST-Net
    - 特征图：Yolact SOLO-v2 Maskformer DDF-CVPR'

2-3-1 输入压缩 硬注意力or软注意力

- 软硬结合：部分**token**硬注意力，部分**token**软注意力 （更适用于多标签，语义分割？）
  - 不重要的token用量化分组，处理多种背景信息，离群点分到去除一组
  - EVIT Evovit; FCABert PnpDetr BAT-CVPR'23

2-3-1 输入压缩 重构矩阵

- **重构矩阵：原型低秩序列重构到原始的序列数目(序列重构矩阵)，重构矩阵也可以理解为上采样矩阵， (N,M)的注意力矩阵 @ (M, D)的压缩序列矩阵**
  - **提取判别性的局部的+高层语义的特征；避免多个position产生相同的特征（过平滑）**
  - **静态**
    - **Bilinear Nearest**
      - sample-inter-ECCV'20 Anytime-ICLR'22
    - 参数化上采样层：不再是Bilinear插值: 单个像素特征可以产生多个像素的特征
      - 特征降维：SIT
      - 快速优化：Dusampling-CVPR'19
      - 序列维度扩张（序列注意力）： Deconv
      - 通道维度扩张（1x1 conv 升维到h*w倍）：Swinunet Pixelshuffle-CVPR'16；升维度操作计算量巨大
  - **动态**
    - 硬注意力：
      - 最近邻量化重构 Cluster-Former ACT Token-Pooling Evovit Tcformer DGE
        - 重构Softmax的权重： Token-Pooling ACT TOME-ICLR'23
    - 软注意力：
      - 特征降维：A2Net APC Pnpdetr
      - 聚类成分：高斯混合聚类，多个类中心加权得到恢复的token：
      - 动态局部插值（每个位置动态产生不同的插值权重）：可以保持每个token的原始位置，适用于swin的压缩，避免了全局无关信息的干扰
        - EDPT-NIPS'22
        - Carafe-ICCV'19 Regproxy
      - 原型和原始序列矩阵乘得到注意力：OCR
      - 原始序列直接作为注意力，特征维度做注意力：ACF
    - 软硬结合：
      - 序列剪枝保留的token硬注意力恢复，剪枝的token软注意力恢复
      - 剪枝token用邻域位置token恢复：Anytime-ICLR'22
  - 跨层维度对齐+跨层注意力：**进一步融合低层次的+局部的特征，辅助局部的+高层语义的特征的提取.(regproxy)**
    - 静态：GUM-BMVC'18
    - 动态：VT-ICCV'21 NRD-NIPS'21 RegProxy-CVPR'22 FADE-ECCV'22

2-3-1 输入压缩 重构矩阵

- 重构矩阵：
  - 并行+重构矩阵：
    - 方式1：重构剪枝层全输出+最后一层压缩输出 Evovit，在输入向量上没有任何重构误差
    - 方式2：重构最后一层压缩输出，Tcformer
    - 方式3：重构剪枝层部分输出+最后一层压缩输出 ：Lengthadaptive-ACL’21 STP
  - 串行+重构矩阵：
    - 每个stage直接重构最后的输出：Segformer
    - stage到stage逐渐重构：FPN Setr Tcformer
  - 层维度+重构矩阵：
    - 重构每个stage/模块的输出：DGE Evovit
    - 重构整个模型的输出(用于密集预测)：Segformer Tcformer
  - 特征维度+重构矩阵:
    - GhostNet：轻量级重构模块

先选择一些Token加权作为类中心，然后量化

N

N'

1/2	**0**	**1/2**	**0**	**0**	**0**
0	1/2	0	0	1/2	0
0	0	0	1/2	0	1/2

基于量化的压缩 Q左乘Q选择矩阵和Q重构矩阵

Cluster attn是一种特例：最左边的矩阵聚类成分

smyrf也是特例：最左边的矩阵one-hot

Query 选择矩阵，行选择矩阵

N

N'

1	**0**	**0**	**0**	**0**	**0**
0	1	0	0	0	0
0	0	0	1	0	0

D

N

in:X

多种压缩组合

$N \times D -> N' \times D$

D

N'

N

N'

0.1	**0.1**	**0.1**	**0.2**	**0.2**	**0.3**
0.1	0.2	0.3	0.1	0.2	0.1
0.1	0.1	0.2	0.2	0.2	0.2

In: 也可以是字典，见压缩总结ppt，这样就整合了剪枝、矩阵分解和权重共享

Query 重构矩阵

N'

N

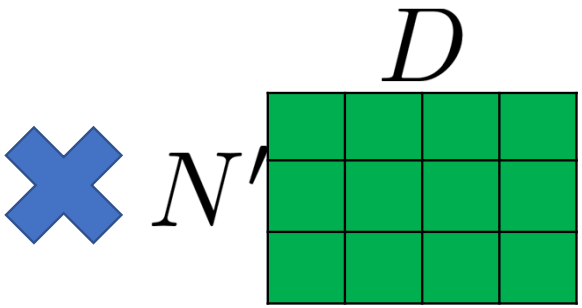
1	**0**	**0**
0	0	0
0	1	0
0	0	0
0	0	0
0	0	1

SMYRF
多个选择矩阵
和重构矩阵，加权平均

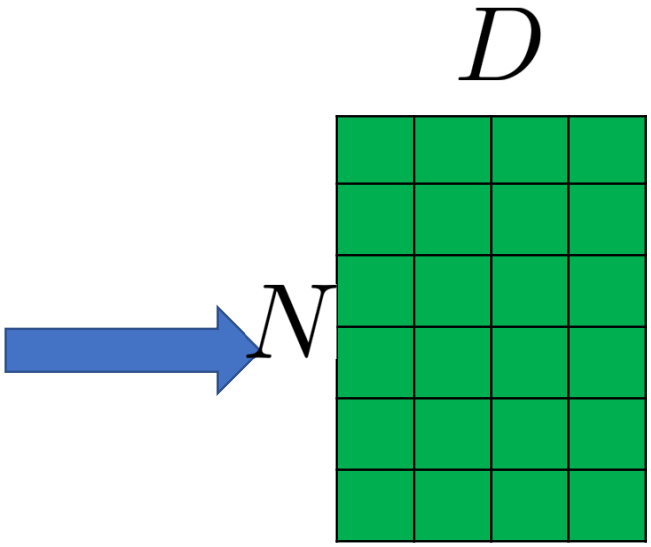
1	**0**	**0**
0	0	1
0	1	0
0	1	0
1	0	0
0	0	1

Cluster att

Query 重构矩阵并不是每行都有1，
且和前面的query选择矩阵对应



固定参数or
选择矩阵得到的结果



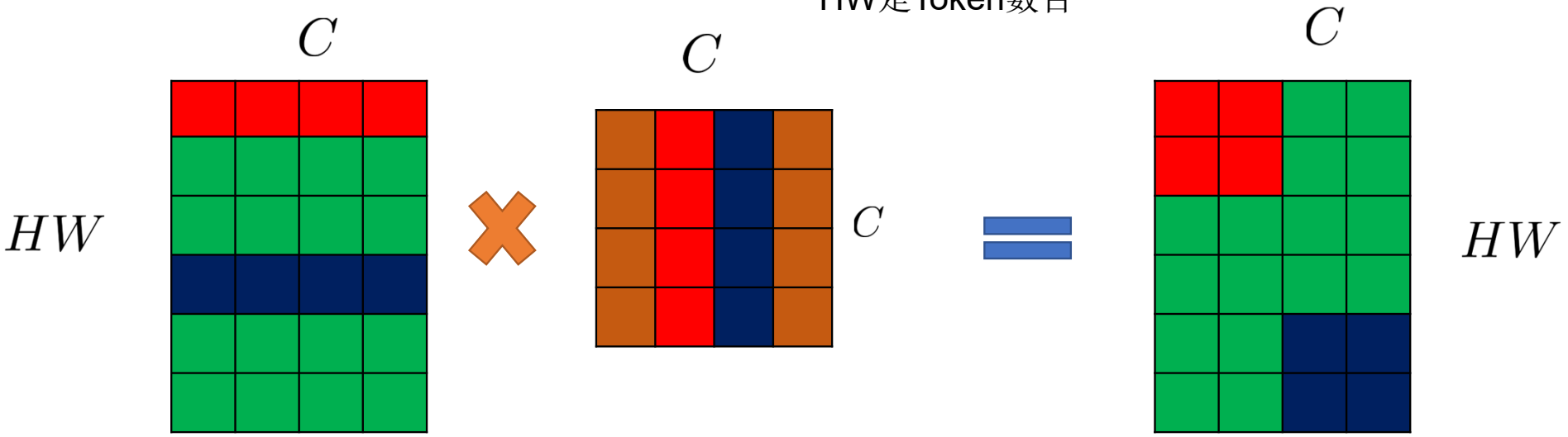
2-3-2 - 输出压缩

- 分块稀疏/分块检索: 根据Q块和K块的相关性，分配每个块rank(包括不规则稀疏)
  - **选择矩阵是{0,1}的硬注意力矩阵，选择输出某些块计算**
  - 这里每个Q块的检索方法相同（不相同的见并行）
- 输出低秩：很多输出位置由很少的输出位置的结果得到：
  - 以SVD为例：**选择矩阵是[0,1]的软注意力矩阵，选择矩阵@原矩阵=压缩结果（或者压缩结果是固定的一组参数作为字典），重构矩阵@压缩结果=原输出矩阵重构结果**
  - 矩阵分解：Kroc-attn Nystrom SOFT
  - 相似QK块检索：相似的QK块用其他QK块的计算结果表示？
- 解码模型-序列化标签：query只能查找排在他们前面的key

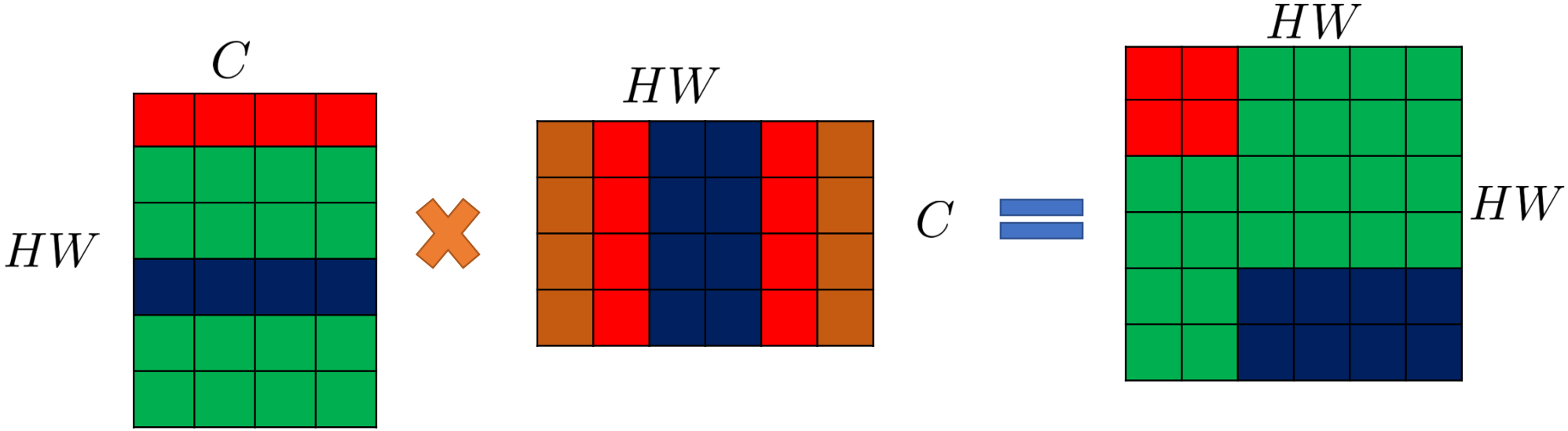
2-3-2 - 输出压缩

红色为样本1的去除位置，
蓝色为样本2的去除位置
HW是Token数目

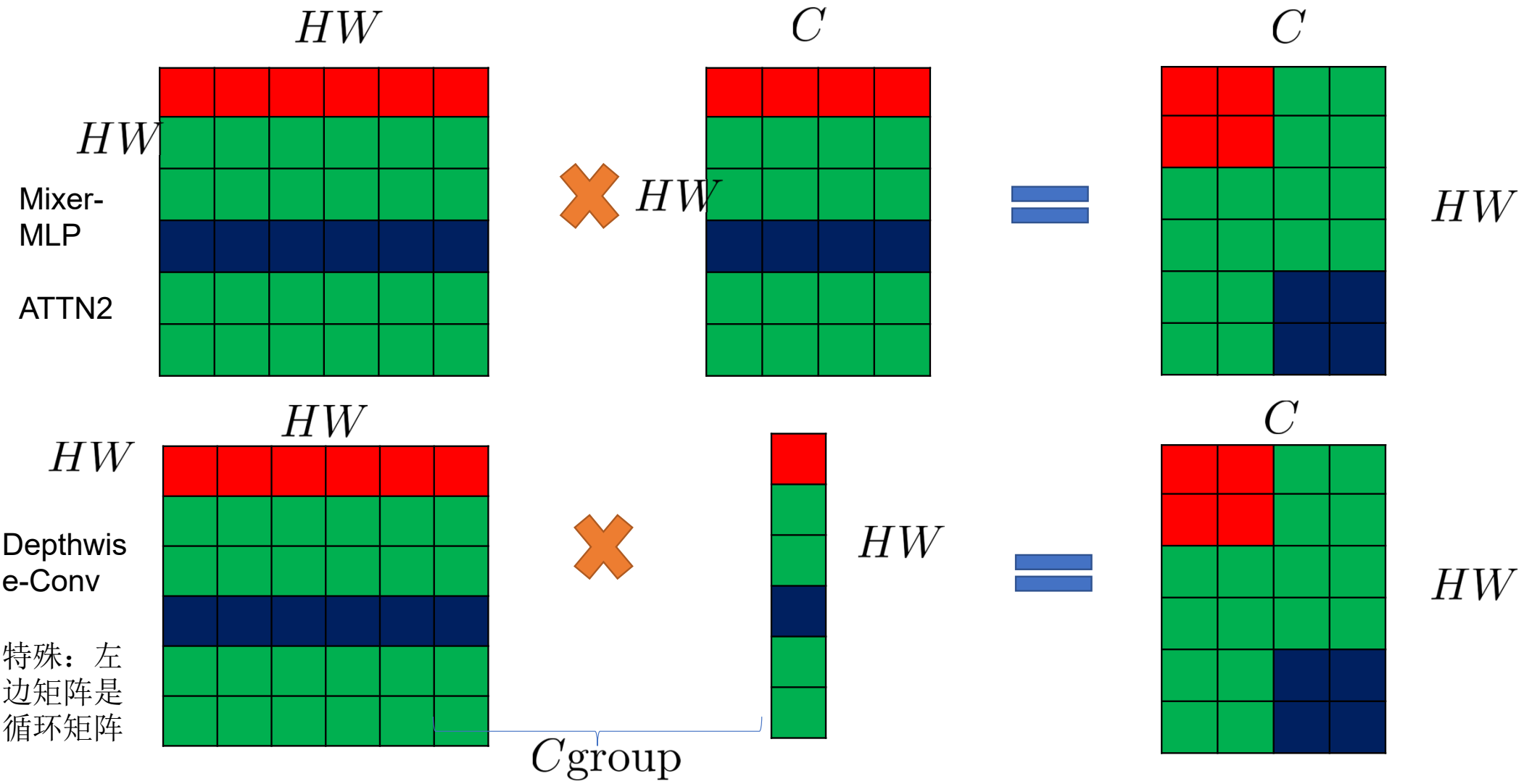
FFN



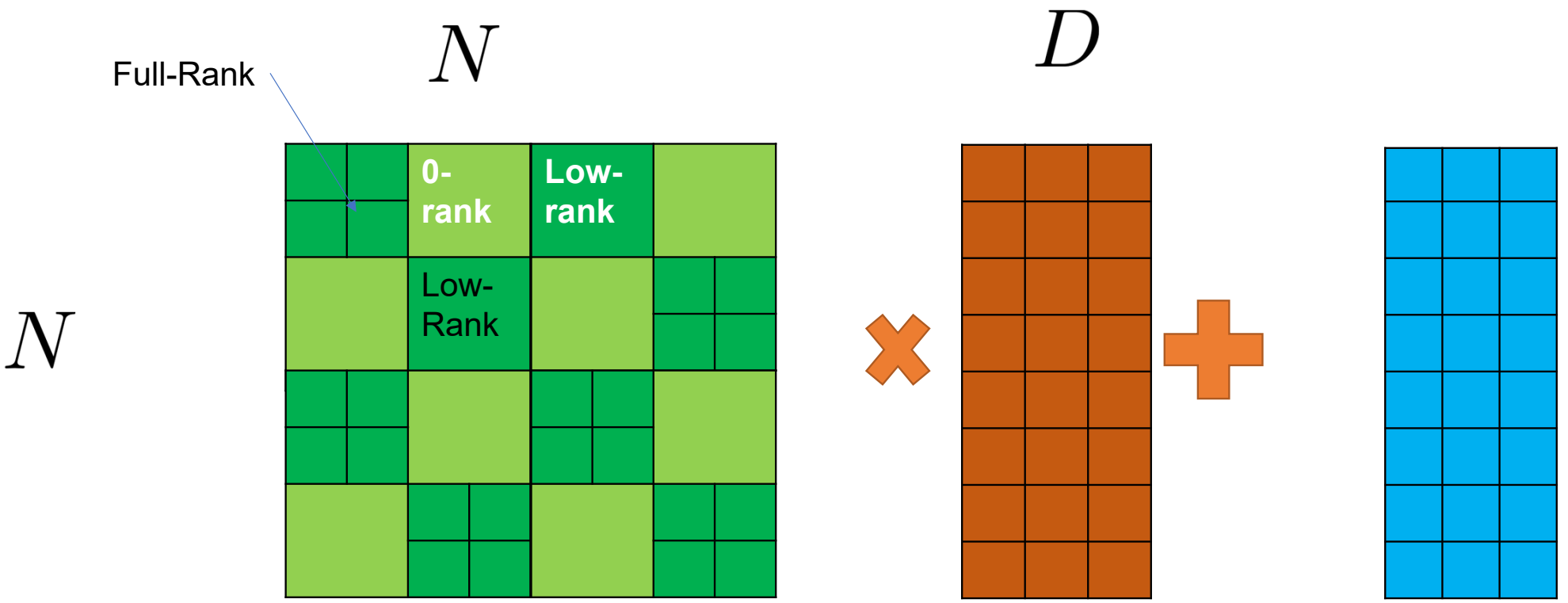
ATTN1



2-3-2 - 输出压缩


HW
C
C
HW
HW
HW
Mixer-MLP
ATTN2
HW
HW
C
HW
HW
Depthwise-Conv
特殊：左边矩阵是循环矩阵
Cgroup

2-3-2 - 输出压缩



Full-Rank 的块是每个位置都要计算注意力；
Low-Rank 的块是一个块计算一个注意力：由组内压缩方法得到低秩的Q，K，从而得到低秩的内积
0-Rank 的块是不计算的位置

2-3-2 - 输出压缩 分组检索

- 基于先验的检索，局部检索可以去除特征中的噪音、无关信息；局部检索提供了先验信息，加速模型收敛
  - 密集注意力：
    - Non-local Transformer
    - 部分Q块检索所有K块，其余Q块采用其他检索方法
  - 块对角稀疏：对角块full-rank + 其余 rank-0
    - 纯对角-组内聚合：FLA SE-net CBAM Swin VAN
    - 分块对角：先验分组下，每个Q组只检索空间位置近的K组：
      - Sparse-Transformer local-attn Bigbird Focal-Transformer
    - 局部连续稀疏：每个Q组定位到一个K组，及其相邻的位置：
      - 动态权重： SASA-NIPS'19 Adaptive-Span dynamic-attn Dynamic-DW-ICLR'22 NAT-CVPR'23 SimVIT Slideformer-CVPR'23
      - 静态权重，卷积：
        - +压缩率设置：RepLKNet，更大的局部窗口
    - 空洞对角：SparseBERT Sparse-transformer longformer
    - 问题：类似卷积，直接使用unfold实现很耗内存（e.g. Simvit SASA）
      - 共享的Q：QnA，减少内存消耗

2-3-2 - 输出压缩 分组检索

- 基于重要性的检索
  - 熵：SIMAM GCT，每个neuron与其他neuron的相似度
  - 最小重构误差：
    - 分组检索：Reformer Smyrf SCT-NIPS'21 NLSA KIT-CVPR'22 Routing Cluster-attn-阶段二 DGT Sinkhorn
    - Token检索：DRS YOSO-ICML'21 ELSA非规则：每个Q检索hamming距离近的K，同时乘上K的范数 KVT-ECCV'22
    - 重构加权误差，每个token给予不同的权重（如cls attn）：Cluster-Attn的直接改进
  - 最小loss：
    - 可学习位置：DCN Deformable-DETR-ICLR'21

2-3-2 - 输出压缩 对角注意力

Squeeze：X 输入压缩（减少计算量）X_c (但是输入压缩会丢失信息，尤其是压缩到1维的情况)

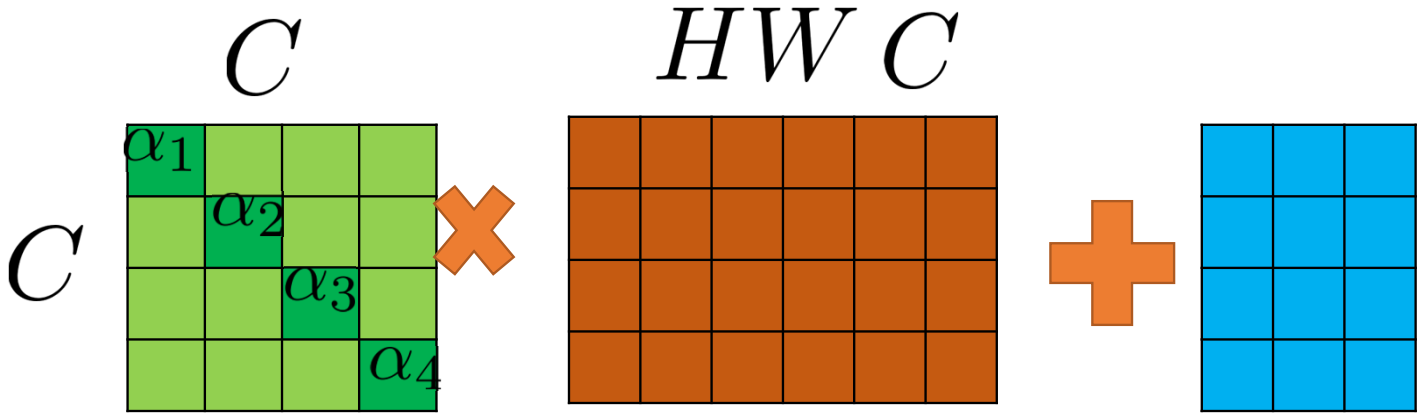
->Excitation：压缩向量自注意力 A_c

->Scale：就是对角注意力 A_c恢复到X大小然后Scale到X上得到A_x

- 通道Scale
  - Sequeeze序列压缩到1维+Excitation通道全检索：SE-Net GSoP CenterMask-CVPR'20
  - 序列压缩到1维+通道局部检索：ECA-CVPR'20 GCT
- 序列Scale
  - Sequeeze通道压缩到1维+ Excitation序列局部检索：BAM CBAM
- 序列+对角Scale
  - Sequeeze序列压缩，压缩到序列H'W' + Excitation序列局部检索：GENet
  - Sequeeze水平序列压缩到1and垂直序列压缩到1+ Excitation通道全检索：CANet

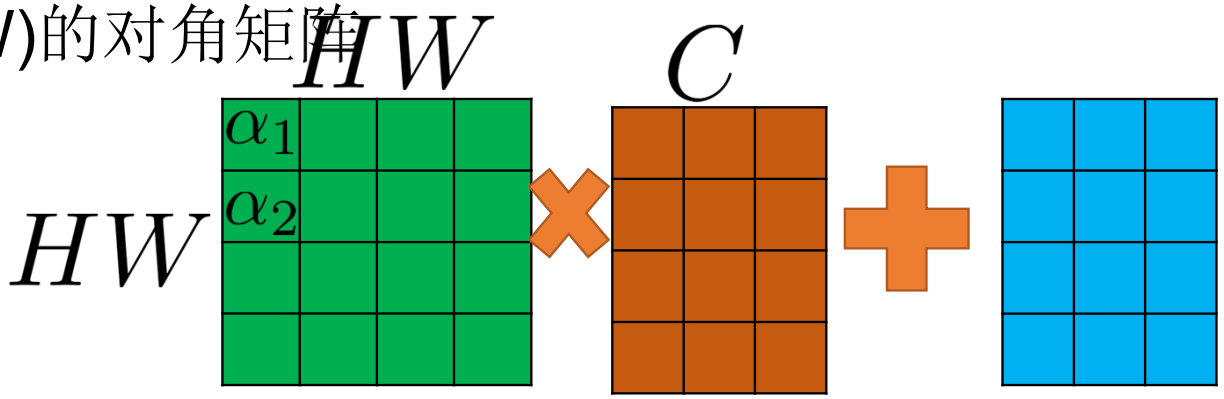
2-3-2 - 输出压缩 对角注意力

- FLA :每个token长度为1, 每个分块长度为C, 每个块内都是CxC的满秩矩阵
- Trip-attn: SE-Net + CBAM



2-3-2 - 输出压缩 密集注意力

- CGNL: 每个token长度为C/G * HW
- DA-Net GNL
- C3-net每个token长度为HW
- CAA：水平注意力+垂直注意力
- Nonlocal CBAM DA-Net Xcit: 每个token长度为C，注意力是(HW)x(HW)的对角矩阵



2-3-2 - 输出压缩 对角注意力

- Se-net:
  - Step1: C/2xC (dense权重，注意力) @ Cx1 (X的pool降维-V) -> C/2x1
  - Step2: CxC/2 (dense权重，注意力) @ C/2x1 -> Cx1
  - Step3: diag(C) (注意力矩阵) @ X (V) 对角注意力
- CBAM
  - Step1: (HW)x(HW) (depthwise权重，注意力) @ (HW)x1 (X的pool降维-V) -> (HW)x1 固定注意力
  - Step1的结果只能捕获局部位置信息，且没有利用通道信息，CA-Net解决
  - Step2: diag(HW) (注意力矩阵) @ X (V) 对角注意力
- SimAM 每个channel单独做attn
  - Step1: diag(HW) (注意力矩阵) @ HWx1 (X的pool降维 V) -> (HW)x1
  - Step2: diag(HW) (注意力矩阵) @ X (V) 对角注意力

2-3-2 - 输出压缩 Q-K检索任务设置

	GPU计算速度	计算量	近似能力
2-3-2 输出压缩	中	中	强
2-3-1 输入不规则稀疏	慢	小	强
2-3-1 输入分块稀疏	慢	中	中
2-3-1 输入行列稀疏（序列剪枝，数据库剪枝）	快	中	中

2-3-3 内积计算压缩

- 循环矩阵乘法（卷积）
  - 卷积->FFT+点乘：CirCNN CirConv GF-NET F-NET
  - https://zhuanlan.zhihu.com/p/176935055
- 部分和代替整体和计算
- 核函数近似
  - 内积计算拆成两个小矩阵Q’ 和K’，近似原来的exp kernel，先计算K’V形成anchor个d维度的anchor，再计算和Q，anchor根据当前的输入K动态决定而不是固定的
    - Linear SGH方法和Linear其实很类似
    - Performer CGNL ，理论误差较大
    - DPFP：序列方法，速度慢
  - 增强注意力矩阵的判别性/区分性
    - 注意力集中部分位置：RFA Cosformer Ripple
    - 兼容位置编码：SPE PRF Permuteformer RPE Ripple-ICML’22
    - +序列对角注意力：Flowformer-ICML’22
  - 核函数作用在KV的内积上：XCIT
  - 多组核函数：LRAR-ICML’22
- 多组内积，计算顺序：
  - 多个内积计算，预先合并，重参数化：EL-ATTN Ac-NET RepVGG
  - 多个内积计算，先计算输出矩阵小的：Performer
  - 投影矩阵和Q/K合并：EL-Attn LISA
  - 层重排

2-3-3 内积计算压缩

- 核函数近似
  - q和anchor的距离矩阵和k和anchor的距离矩阵的乘积作为近似的相似度矩阵：AGH
  - 核内积近似细粒度到每个组内
  - 减小K的序列长度，进一步加速
  - 动态锚点数目的performer，加gate

2-5 静态压缩：固定参数

固定参数可以编码一些固定的先验信息：如相对位置信息，类别/单词共现信息；同时可以编码整个数据集的一些关键信息

- 固定的注意力数值，无需计算（静态+输出选择矩阵）
  - 全局权重作为attention: Mixer-MLP gMLP GF-NET AFNO
  - 局部权重作为attention，局部信息增强：Container ConVIT Uniformer ConvNext
  - 无参数化：Poolformer
- 固定的K/V，无需计算，可以任意长度
  - anchor作为全局参数，K（输入） 直接作为参数：External-attn
- 固定的Q
  - 全局检索：Set Transformer LUNA-nips'21 NextVIT SIT ABC'ACL'22 LightVit GPVit-ICLR'23
  - 局部检索：QNA-cvpr'22
- 固定的统计量：加速归一化模块的计算
  - BN UN-MM'22

2-5 静态压缩：固定参数

固定的注意力数值：位置编码

- 绝对位置编码
  - 纯学习的：BERT，没有外推性 Medt-MICCAI'21
  - 正余弦编码：Transformer，提供了相对位置信息
  - 每个Q学习一个位置参数：Localness Adapative-Span Dynamic-attn
  - 固定的位置参数：Hard-Coded
- 相对位置编码：
  - 在QK.T加上相对位置编码，等价于在exp(QK.T)乘上一个重要性得分，可以用于增强局部信息；如果得分为0，那么等价于分块检索的压缩
  - 位置编码只依赖相对位置，不依赖绝对位置，相当于增加归纳偏好:平移不变性:Y，翻转不变性？
  - 相对位置数目作为参数数目
    - T5 RPE TUPE Deberta Swin
    - DMAN
  - 相对位置编码纯先验：
    - Transformer ContextPool PerVIT-NIPS'22
- 先验性质
  - 单调性、对称性、等变性： APE-RPE

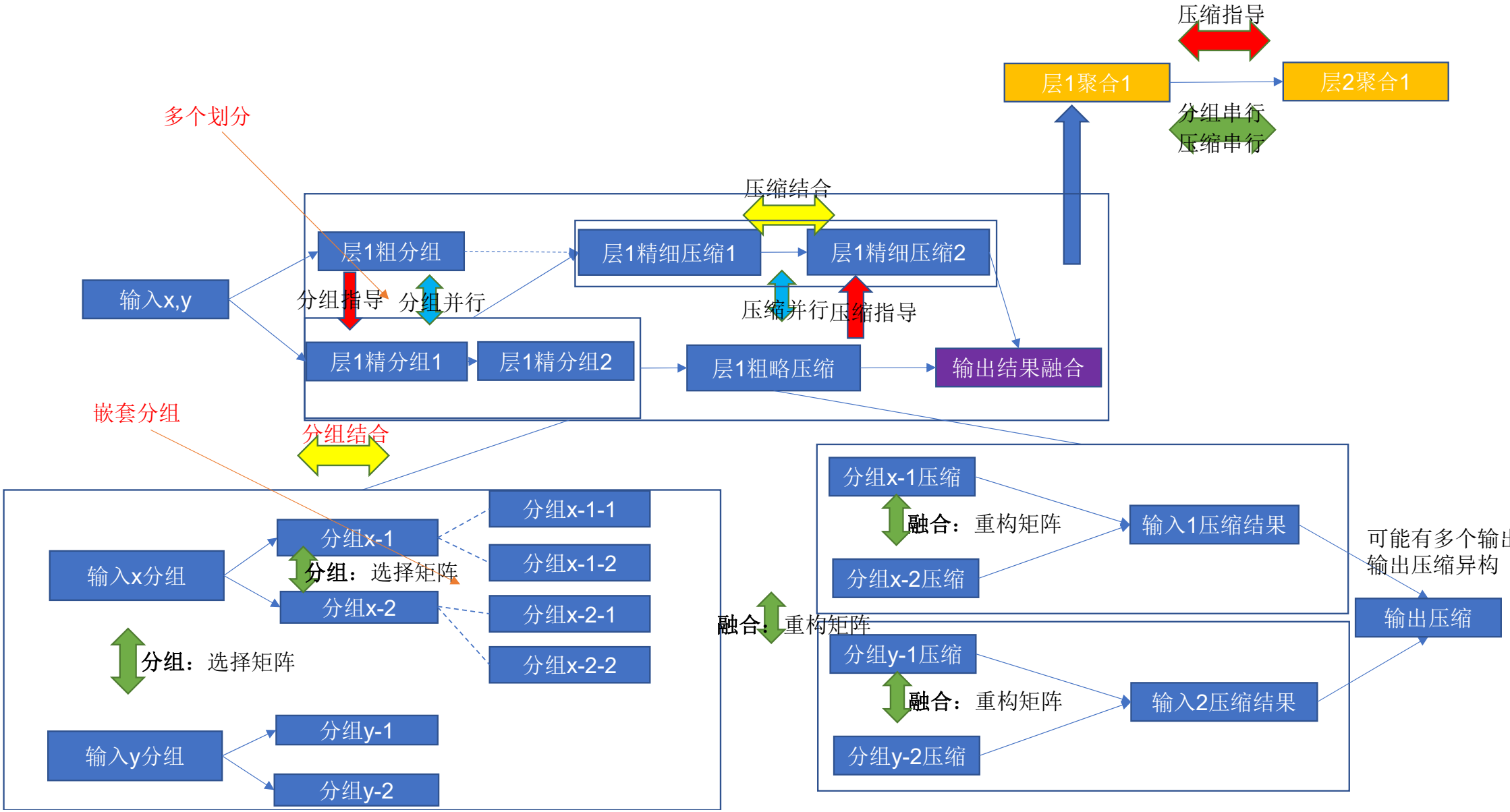
2-5 静态压缩：固定参数

- 固定参数-异构：不同分组采用不同的固定参数
  - GhostNet Mixconv Hetconv GoogleNet: 参数矩阵当作输入,$W \in R^{N_{in} \times N_{out} K^2}$ 对输出维度进行分组，有的组是1x1卷积，有的组是3x3卷积

3 压缩组合

组合方法：不同模块不同的压缩方法，每个模块在尽可能高的压缩率下，得到不同角度的**表达能力**，每个分支只专注提取各自分支的信息。

- 组合方法：
  - 3-1 并行
  - 3-2 串行
  - 3-3 折中
  - 3-4 结合
  - 并行vs串行vs结合：
    - 串行在每层都会忽略一种信息；串行后面的层会覆盖前面提取的特征
    - 并行可以在一层同时捕获多种信息；并行提取不同的特征，在多分支聚合时会导致冲突
    - 结合可以避免上面问题
- 交互方法：
  - 3-5 交互位置
  - 3-6 指导
  - 3-7 融合


压缩指导
层1聚合1
层2聚合1
分组串行
压缩串行
多个划分
压缩结合
层1粗分组
层1精细压缩1
层1精细压缩2
输入x,y
分组指导
分组并行
压缩并行
压缩指导
层1精分组1
层1精分组2
层1粗略压缩
输出结果融合
嵌套分组
分组结合
分组x-1压缩
融合：重构矩阵
输入1压缩结果
分组x-2压缩
可能有多个输出
输出压缩异构
分组x-1-1
分组x-1
输入x分组
分组：选择矩阵
分组x-2
分组x-1-2
分组x-2-1
分组x-2-2
融合：重构矩阵
输出压缩
分组：选择矩阵
分组y-1压缩
融合：重构矩阵
输入2压缩结果
输入y分组
分组y-1
分组y-2
分组y-2压缩

3 压缩组合

- 3-1 并行
  - 划分并行：每个划分采用不同的压缩方法 -> 类似集成
  - 分组并行：每个分组采用不同的压缩方法 ->
    - e.g. 多个head采用不同的压缩方法，
    - e.g. 部分序列采用序列压缩、部分序列采用分组检索；
    - e.g. 部分序列采用动态选择、部分序列采用静态选择;
  - 多任务并行
  - 多设备并行

3-1 并行：同一划分/分组多种压缩

- 划分并行+多维度划分并行：（捕获多维度各自的特征，从而提取更有判别性的特征）
  - 水平+竖直+2d空间： Stripe-Pooling-CVPR'20 Axial-ECCV'20 Hire-MLP SparseMLP VIP Medt-MICCAI'21 CA-Net-CVPR'21 Seaformer-ICLR'23
  - 序列+特征： SAMNet-TIP'21 Wave-MLP-CVPR'22 STD-ICML'22 HAT-CVPR'23
  - 序列+时间： MTV MLP-3D M3T-CVPR'22
- 多分组并行：同一输入产生多个分组，每个分组是原输入的一个完整划分（可以是多个相同的划分，也可以是多个不同的划分，）
  - Hit GG: local + strided
  - 多尺度分组：可以有重叠分组，最终分辨率相同
    - OCNet Asym-nonlocal-ICCV'19 LG-Trans Shunted-CVPR'22 QuandTree-ICLR'22 MPVIT-CVPR'22 HIPT ScaleFormer
    - 基于注意力的融合：Focal-Transformer IPE DWN
  - 多种分组方法：
    - SLIC分组+Grid分组：CAST SMESwin

3-1 并行：同一划分/分组多种压缩

- 划分并行+压缩方法：多种选择矩阵，多种表达能力并行
  - 多种输入压缩：
    - 多种数据增强：COSOC-NIPS'21 GF-Net-NIPS'20 AnchorNet-ICME'22
    - 多种裁剪位置： GF-Net-Nips'20，优势：捕获关键位置/部位，处理空白很多的图片
    - NTS-CVPR'21
  - 多种输出压缩
    - 多感受野特征：
      - FocalModule-NIPS'22 MixConv SK-Net Inception BCDNet-ECCV'22 TridentNet-ICCV'19 LVT-CVPR'22 SegNext-NIPS'22
    - 密集检索+局部检索：
      - CoupleNet ASPP Container-NIPS'21 AC-MIX-CVPR'22 Conformer Mobileformer-CVPR'22 Transformer-XL ELSA DebertA Nommer-CVPR'22 HR-NAS DFVT-ECCV'22
    - 密集检索+标签特定检索：IS-Net-ICCV'21
    - 密集检索+自检索：VIT残差边 AUG -S Restv2 Mvit MobileViTv3 Missformer GhostNet
      - 增加信息流通性，利于训练深层网络
    - 局部检索+自检索：ResNet DenseNet CliqueNet
    - 密集+局部+自检索：Ada-connect-CVPR2019

3-1 并行：同一划分/分组多种压缩

- 划分并行+压缩向量
  - 输入压缩+输出压缩：
    - 输入压缩+分组检索：
      - 序列维度：Acontext-ICCV'19 LS-Transformers Scatterbrain GL-Net-ICCV'19
      - 特征维度：多组特征
        - DASA 组内full-attention，组间特征压缩到1维度-对应attetnion
    - 多尺度分组->密集检索+分组检索：Focal-Transformer
  - 内积压缩+输出检索：
    - 局部检索：Scatter-brain Combiner FMM Groupformer LHA
    - 对角检索：Flowformer
- 划分并行+分支门控（集成）
  - 参数门控：Diverse-VIT
  - 模块门控：FPN->AQD

3-1 并行：同一划分/分组多种压缩

- 划分并行+固定参数：
  - 静态参数+动态输入：Bievolution-AAAI’22 DINO-ICLR’23
- 划分并行+计算量分配：
  - 多种压缩率/分辨率/多种裁剪位置：Stride卷积 Pooling
    - 输入层：
      - 多种分辨率： RA-Net-CVPR’20 Elastic-CVPR’19 Attn-Scale DR-Net IC-NET RefineNet SwiftNet-CVPR’19
      - ClusTR-2022
    - 中间层：Bisenet HR-NET-CVPR’19 MDP-CVPR’20 DDRNet-2021 Hrformer Dynamic-Routing LitePose-CVPR’22 HR-VIT-CVPR’22 STP AFF-WACV’21 PIDNet-CVPR’23 Potter-CVPR’23
  - 多种层：BiseNet

3-1 并行：同一输入/分组多种输出

- 多分组并行
  - 嵌套分组，维度分组之后部分采用先验分组，部分采用量化分组：Reformer SMYRF
- 分组并行+计算量分配
  - 每组压缩率不同，但是不同样本的压缩率配置都相同，比如K组不压缩，N-K组压缩到K’
    - 不压缩+压缩：
      - Evit EvoVIT
    - 每个分组自适应的压缩率：
      - 多输入分组+压缩率：ADSH DAPH
      - 序列分组+压缩率：
        - 局部检索+全局检索(不压缩): Universial-attn Longformer Bigbird ETC ViL Star-transformer
        - 每个Q组检索不同数目的K组：Sanger DSA ELSA ISCA’2021 YOSO ZAP
        - 序列数不同：DGE（每个region为单位看待） Segblock CTS-CVPR’23
        - 层数不同：
          - PointRend-CVPR’20 MagNet-CVPR’21 Dvit-v2 HPPU-SCIS’22
          - TTC：部分维度用adapter部分不用
      - 特征分组+压缩率：
        - 全局检索+局部检索(不压缩): Lite-Transformer-ICLR’20 DS-Net-NIPS’21
    - 固定参数压缩：UVC Tile-Sparse BLK-REW

3-1 并行：同一输入/分组多种输出

- 分组并行+压缩方法：多种选择矩阵
  - 多种矩阵压缩：Pixelfly Butterfly DeBut
  - 序列分组+压缩方法
  - 特征分组+压缩方法
    - ShuffleNetv2
    - 多种输出检索：
      - 局部检索+间隔检索：MaxIM-ECCV'22
      - 局部检索+自检索：DenseNet ShuffleNetv2 Faster-Net-CVPR'23
    - 多种输入压缩：
      - Maxpool+avgpool选择矩阵：Iformer-NIPS'22
  - 每组的稀疏模式都不同:
    - N:M Sparsity Falcon
- 分组并行+压缩向量
  - 维度分组：
    - 输入压缩+输出压缩： HiLo-NIPS'22

3-1 并行：同一输入/分组多种输出

- **多输入异构压缩: e.g. Q,K异构压缩**
  - 多种压缩方法：
    - Q,K采用相同量化中心 or Q,K采用不相同量化中心 ： Routing
  - 多种压缩率：
    - Q,K采用不同的分组大小
    - 只压缩QK：SepVIT
    - 只对KV压缩：linformer PVT CoCs-ICLR'23
    - 只对Q压缩，Cluster-attn阶段一
  - 序列到序列模型，编码-解码
  - Mixformer 模板和当前图像检索不同的patch

3-1 并行：同一输入/分组多种输出

- 分支位置：分支早可以提取更有判别性的特征，分支位置晚可以减少计算量，分支位置早可以每个分支提取更有信息量的特征
  - RCNN系列每个ROI可以认为一个分支
  - 输入层：
    - 多输入数据：
      - 多模态，e.g.多模态数据在输入的时候直接区分开：DCMH
      - 多划分：MPVit Cross-Vit IC-Net FCB-2023 Merit-MIDL'23
  - 中间层：
    - 多输出标签：Faster-RCNN DoubleHead-CVPR'20 TSD-CVPR'20
    - 多输入数据
      - 多划分：HR-Net MPVit
  - 最后一层：
    - 多输出标签：PARSE-Net-ICLR'16 R-FCN-NIPS'16
    - 多输入数据
      - 多划分：VITDet-ECCV'22 YOLOF UVIT Higher-HR-CVPR'20
      - STDN-CVPR'18

3-1 并行：同一输入/分组多种输出

- 分组并行+分支门控，e.g分组特定Token
  - 序列门控：额外的聚类token：NCM-ACL'22 GMPool
  - 序列门控：窗口特定token：MSG-CVPR'22 RegionVit Sepvit FasterVIT

3-1 并行：同一输入/分组多种输出

多任务，任务并行，模型共享

- 分支门控：多分支，任务共享部分分组，每个任务使用部分分支(0个/1个/多个)，每个任务包含任务特定的分支，增强各个任务的建模能力：
  - 模块门控（包含通道门控）：每个任务使用总模型的部分模块
    - 单输入-多任务输出：
      - 分类+回归框：Decouple-Head-CVPR'2020 YOLOX
      - 密集分类+整体分类：EncNet LVVIT HCE SANet
      - 分割+检测：BlitzNet NDDR Disparse
      - 分割+深度：PAP-CVPR'19 DeMT-AAAI'23
    - 多任务输入-多任务输出：
      - 与Backbone并行：Compactor，无法节省训练显存
      - 与Backbone串行：节省训练显存
    - 单层adapter，可以合并到参数里：LoRA Fedlora-ICLR'22 Kadaption-AAAI'23 Fact-AAAI'23 IA3-NIPS'22 Adalora-ICLR'23 Dylora
    - 多层adapter，不能合并到参数里：Adapter-ICML'19 Adaptformer-NIPS'22 VL-Adapter-CVPR'22
    - 并行：Lora PETL-ICLR'22 并行结果更好？
    - 串行：adapter Readapter
  - 索引门控（矩阵分解）：HybirdGrid-ICLR'2021 Compactor-NIPS'21 Atom-ICLR'22
  - 参数门控（每组参数视为一个分支）：
    - PackNet Piggyback-ECCV'18 SupSup DEN-ICLR'18
    - TinyTL SparseShare Ptuning-ACL'21 Ptuningv2-ACL'22 SSF-NIPS'22
  - 序列门控（每组序列视为一个分支）：
    - 分类：L2P VPT-ECCV'22 Prefix-ACL'21 Dytox Meat
    - 分割：HQ-SAM Oneformer-CVPR'23
  - 组合：
    - 序列门控+模块门控：UniPELT-ACL'22 NOAH-有的层可以没有任何adaptor
    - 模块门控+参数门控：AANet-CVPR'21 DER-CVPR'21
    - 索引门控+模块门控：vit_adaptor

3-1 并行：同一输入/分组多种输出

- 共享方式：每个分支可以所有任务共享/几个任务共享/任务独享/0任务共享（剪枝）
  - 只共享Pretrain:
    - Adaptor-CVPR'18 Nettailor-CVPR'19 Adashare TAPS-CVPR'22 ConvPass LST-NIPS'22
    - LoRA-NIPS'21 Fact-AAAI'23
  - 只共享Pretrain+一个新模块：Hyperformer-ACL'21 Polyhistor-NIPS'22
    - 有利于多任务相互促进
  - 部分共享：相似任务共享参数 e.g.相似任务聚类 相似任务检索
    - PNN PackNet Piggyback Atom
    - 部分共享+0任务共享：CPG
  - 全共享，软门控：Cross-Stitch MOE M3VIT Adaptfusion-ACL'21
  - Pretrain参数也参与微调导致预训练知识的遗忘？
- 样本依赖
  - 样本自适应分支模块选择/prompt选择：DAM-VP-CVPR'23
  - 样本自适应分支模块生成/prompt生成：vitabs-CVPR'23

3-1 并行：同一输入/分组多种输出

- 多任务关系建模？
  - 相关性高的任务，输出结果要一致：分类和回归框的结果要一致，要么置信度都跟高，要么都很低
- 多设备：见应用任务
  - SuperVit
- 多机器：联邦学习 （多机器）
  - 机器独立参数，适用Non-iid：FDL FEDPHP
- 多域：
  - 域特定分支：DSBN-CVPR’19

3-1 并行：同一输入/分组多种输出

- 多输出标签：多输出并行分支，增强特征的判别性
  - 序列门控：额外固定序列： （类似cls-token）
    - Aug-SA MESH-单词共现信息 ST-VIT
    - 类别特定token/query（类别特定token作为Q，特征token作为K）：
      - 交叉注意力： C-tran CSRA-ICCV'21 Q2L MCTformer-CVPR'22 MMCAP Token
      - 自注意力+交叉注意力： ML-GCN-CVPR'19 SSGRL-ICCV'19 Tokenpose-ICCV'21 Segmentor-ICCV'21 SegVit-NIPS'22
        - 引入自注意力考虑了类别之间的关系
      - 匹配方式：
        - 静态-小于类别数：ML-decoder-WACV'23
        - 静态-大于类别数：MCIBI-ICCV'21 ProtoSeg-CVPR'22
        - 动态： Maskformer-NIPS'21
      - 粗分类初始：ACF OCR PCAA-CVPR'22
    - 实例特定token/query：
      - Detr Maskformer K-Net
      - TokenPose PPT
  - 模块门控：类别特定的模块：每个分支，只捕获和当前类别相关的区域：
    - 类别特定特征/通道：ACF-ICCV'19 OCR-ECCV'20 PCAA-CVPR'22
    - 类别特定分支：LSTM-ATTN-ICCV'17 SRN-CVPR'17 CDGC IS-Net RECAM
    - 类别不平衡：e.g. 长尾
      - 头部类、尾部类分开的分支：RESLT
- Query特定特征
  - Conditional-DETR DAB-DETR-ICLR'22

3-2 串行：多层采用相同/不同聚合算法

- **模块串行：多模块为单位or单模块为单位**
  - 固定结构：每个模块都是相同的结构
    - 固定的序列压缩方法：固定分组方法：VIT DeiT
      - 多个像素当作整体处理，分块太大，分辨率太低；每个像素平等对待,忽略了一个patch内部的细节信息
    - 优点：
      - 无须设计多尺度特征融合模块
      - 上下游任务可以用相同的backbone
  - 层次结构：每个层次采用不同的聚合算法
    - 浅层小压缩率，深层大压缩率：**(多尺度层次)**
      - 序列压缩：层次ResNet结构，PVT PIT-ICCV'21 HVT-ICCV'21 Swin Hireal-attn
      - (固定)参数压缩：HBP-ECCV'18
      - **考虑多尺度信息，包括底层的细节信息和高层的语义信息；大的分块无法捕获每个块内的细节信息；依赖cls才能捕获高层语义**
      - 适合密集预测任务：多尺度特征，允许高分辨率的分块
    - 浅层局部感受野，深层全局感受野：**（多感受野层次）**
      - TMMC-IJCAI'20 LIT-AAAI'22 Uniformer HIVIT-ICLR'23
      - +多尺度层次： Levit MobileVIT-ICLR'22 Topformer-CVPR'22 EfficientFormer-NIPS'22 EfficientFormerv2
      - Nest 局部检索范围逐渐扩大到全局检索 **（多种感受野）**
    - 多种分支数：HR-Net-CVPR'19
    - 多种压缩率：
      - +自适应层数：MSDNet RA-Net Anytime-ICLR'22 MESS-ECCV'22
      - +自适应序列数目：ATS LTP-KDD'22
      - RTMNet

3-2 串行：多层采用相同/不同聚合算法

- **模块串行：多模块为单位or单模块为单位**
  - 交替结构：不同的聚合算法交替
    - **多种维度聚合交替：**
      - 同时捕获空间关系和通道关系，空间聚合+通道聚合：
        - ELAN-ECCV'22 Davit-ECCV'22 AAU-Net-TMI'22
        - 卷积网络：FFN -> Shift-add
        - 自注意力网络：FFN -> Attn
        - 序列+层次：Dyhead-CVPR'21，层次特征也算在特征维度里
      - 水平+垂直：
        - CC-Net Axial RaftMLP SOTR GhostNetv2-NIPS'22
      - 时序+空间：STTS-ECCV'22 ARN-ECCV'22
      - 神经元合并（输入特征+输出特征） Neuron-Merge RED DCFF
    - 多种分组交替：**（多感受野形状）**
      - 维度分组：ShuffleNet IGCV Sret-ECCV'22
      - 序列分组：
        - 局部分组：Swin Shuffle-Trans
        - 间隔分组：MaxVIT-ECCV'22
    - 多种输入交替：高阶交互，1. 第K层取前K+1个所有输入交互的结果，2. 第K层取第K+1个输入交互的结果 + 当前输入
      - 多个分组输入：Res2Net HorNet-NIPS'22 CPFNet-TMI'20 EDN-TIP'22 CGA-CVPR'23
      - 多个划分：DenseNet DenseASPP
      - 一阶矩（均值）-二阶矩（方差）- 高阶矩：D-TDNN SRM-ICCV'19，解释：二阶矩阵需要依赖一阶矩作为输入
    - 多种压缩向量：
      - 全局检索-输入不压缩 + 输出局部检索：**（多感受野大小）**
        - 卷积：CVT Ceit ContNet Litemono-CVPR'23
      - 全局检索-输入压缩 + 输出局部检索：**（多感受野大小）**
        - 卷积：PVTv2 UT-NET-MICCAI'21 CMT-CVPR'22 WaveMLP RestFormer Seaformer-ICLR'23
        - 窗口注意力：TNT-NIPS'21 Twins-NIPS'21 ScalableVIT-ECCV'22 EdgeVIT-ECCV'22 Dualformer GCViT'22
    - 多种压缩方法：
      - 多种输出检索**（多感受野形状）**:CETNet-ECCV'22
    - 多种压缩率：
      - FFN和MSA采用不同的序列压缩率，UVC-ICLR'22 SAVIT-NIPS'22

3-2 串行：多层采用相同/不同聚合算法

- **模型串行/级联，逐步精细（多阶段）问题？**
  - 多阶段检测：Cascade-RCNN RefineDet HTC
  - 多阶段姿态：Hourglass
  - 多阶段分割：CascadePSP-CVPR'20 Spitio-Recurrent-CVPR'19 CascadeUNet'20 Medsegdiff'22 Uncertainty-cascade-MICCAI'22
  - 多epoch结果级联：FANet
  - 逐渐降低精度：TEL
  - 多种压缩率
    - +动态计算量分配：GF-Net-NIPS'20 DVT-NIPS'21
  - 级联+输入压缩：去除冗余的无关的特征，关注重要位置
    - 不确定特征
      - Urim UACA UAM
    - 粗分类特征
      - RPN Faster-RCNN GBC-CVPR'22
      - PointRend BioNet PraNet Rfnet nnUnet
- 串行顺序：
  - Layer-reorder-ACL'20 bs-conv-CVPR'20
- 串行分支，每层不能同时提取多种特征，表达能力不足

3-2 串行：多层采用相同/不同聚合算法

- 串行任务：最终任务拆分为多个子任务，使用子任务的 foundation model
  - 定位+生成：Inpainting anything
  - 定位+语义：ODISE-CVPR’23
  - 定位+回归：Matting anything

3-3 折中

- 两种模块的中间状态
  - Maxpool, Avgpool -> SoftPool
  - L1-Pruning GM-Pruning -> LFPC-CVPR’21

3-4 结合：多种聚合方法结合

- 分组结合：分组 + 再分组
  - 先验分组+先验再分组：CF-VIT H-trans-1D
- 压缩结合：压缩的基础上在压缩，多个压缩模块整合到一个
  - 多维度压缩：多维度视为整体进行压缩
    - 维度+序列：
      - 对于WX，特征低秩只能压缩W的列，序列压缩只能压缩W的行
      - MDP-CVPR'2020 GLC
      - MIAFormer WDPruning Vit-slim-CVPR'2022 Adavit RKHS-Vit-ICME'22 SAViT-NIPS'22
      - Seamless-AAAI'22
    - 层+序列：TOKEE Anytime
    - 通道+序列：DoP-BMVC'22
    - 序列+层次：Pyramid-conv-CVPR'20 Adamixer-CVPR'22
    - 序列+时间：UX-net-ICLR'23
  - 数据库压缩+序列压缩
  - 多压缩向量：
    - 序列压缩+分组检索
      - 1维输入：Sinkhorn Block-Skim 块对角full-rank ， 其他rank0 ,切除了一些K块 SparseDETR
    - 数据库压缩+ 分组检索
      - Poolingformer APFFormer-TMI'22: 数据库压缩(Pvt)+ 注意力检索(Local)
      - EVA-ICLR'23

3-4 结合：多种聚合方法结合

- 分组，压缩结合
  - Cluster-Prune：量化分组->组内序列低秩：CUP FPKM Tcformer Dense-CLIP
  - Prune-Cluster：重要性分组->组内量化分组->组内序列低秩：FCA-Bert Pnp-DETR

3-5 交互位置

- 多个压缩分支交互方法（交互）
  - 3-5 交互位置：
    - 对称位置：
      - 模块输入： RegionVIT MSG
      - 模块输出（多层之后交互）：
        - 多分支并行：Inception HR-Former Focal DWN IPE MPVIT
        - 多层串行： ResNet DenseNet Aug-s Restv2 DMRNet SEVIT-MICCAI’22 RFANet-CVPR’20
      - 每个级联输出：HTC
      - 网络输出：
        - 模型集成，中间层分支，最后一层融合
        - Cross-VIT， Revit
        - +压缩向量
          - 输出压缩：HIPT
      - 网络输出->网络输入：DVT CF-VIT-AAAI’23 Travelsral

3-6 指导

- 3-9 指导：一个压缩分支指导另外一个分支的压缩，
  - Cluster-Attn Sinkhorn

3-6 指导

- 多分支指导（指导）一个压缩分支指导另外一个分支的压缩，
  - 粗粒度压缩指导细粒度压缩：
    - 特征降维->分组检索：DSA DRS CG-Net
    - 低精度/量化->分组检索：DSA SeerNet DMI PG FracBNN Sanger
    - 序列压缩->分组检索： Cluster-Attn DGT-IJCAI'22 Sinkhorn Biformer-CVPR'23 SparseFiner-CVPR'23
    - 层次指导：Quadtree
  - 这一层的分组依赖上一层的分组结果：上一层的结果可以近似为这一层的结果，
    - 上一层得分最低的几个token，这一层仍然最最低 Evit
    - 上一层得分最低的几个token，直接拷贝到下一层 Evovit
    - 顶层attn不能反映底层原始token的重要性：TransFG
  - 依赖之前多层输出
    - Dvit CP-VIT A-ViT
    - Pabbe-NIPS'20 Skipnet
    - Bnext'22
  - 模型并行：这一层级的模型输出指导下一层级的压缩
    - 细粒度分组依赖粗粒度分组的结果
      - CF-VIT-指导下一层级的再分组
      - RA-NET DVT GF-Net-指导下一层级输入压缩率
      - Blockdrop GaterNet TraversalNET SACM -指导下一层级的输入压缩

3-6 指导

关键是M的计算依赖X,W (或者X、W得到的近似输出) 还是额外的参数E, M = f(X,W, E)

- 压缩依赖：依赖位置 依赖目标 依赖历史
- 指导内容：分组方法 输入压缩 输入压缩 内积压缩

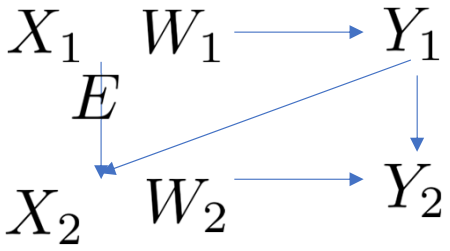
$$X_1 = M_1^x \circ X, M_1^x \in R^n$$
$$W_1 = M_1^w \circ W, M_1^w \in R^n$$
$$Y_1 = X_1 W_1$$

粗略压缩

$$X_2 = M_2^x \circ X, M_2^x \in R^n$$
$$W_2 = M_2^w \circ W, M_2^w \in R^n$$
$$Y_2 = X_2 W_2$$

精细压缩

FBS就是最左边这条线:
先用pool压缩输入得到X_1，借助额外参数E，得到通道压缩的结果X_2



3-6 指导 依赖位置-指导位置

指导/依赖	无任何依赖	上一层输出	多层之前输出	粗粒度压缩得到近似输出	多级模型，上一级的模型输出
分组方法					• CF-VIT
输入获取/压缩	• 随机稀疏（参数or样本维度s）	• PowerBERT Length-adaptive LTP EVit • Real2Bin • Attn-share	• Dvit EVO-VIT Skipnet DFS • BAM • PABEE	• Random-hash • Informer ATS	• Blockdrop QQP GaterNet TraversalNet • DVT GF-Net
输出获取/压缩	Sparse-Attn SMYRF			• Reformer Sinkhorn Cluster-Attn • DSA • CG-NET • SeerNet PG Sanger • LHA	
内积计算压缩	CirCNN GF-NET				

3-6 压缩算法指导-依赖历史

- 当前迭代
  - Slimmings DNS
- 初始化
  - 剪枝-微调算法
  - Lottery
- 中间迭代结果
  - Lottery Rewind

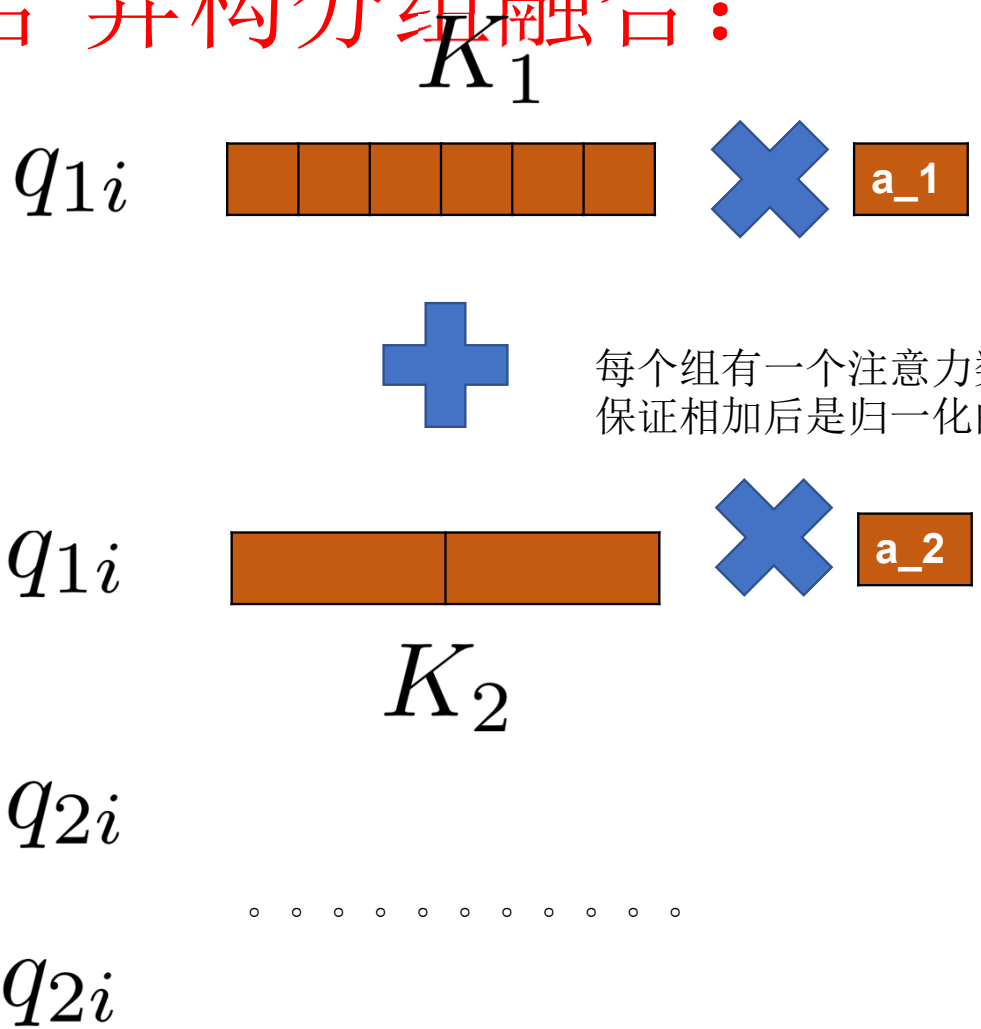
3-6 压缩算法指导-指导内容

- 分组方法
  - 指导再分组方法：CF-VIT
- 输入压缩
  - 压缩率设置
    - DVT 压缩率的粒度是模型层级数目
- 输出压缩
- 内积计算压缩

3-6 压缩算法指导

- 问题0：查询样本裁剪和基于先验的local注意力不兼容：evovit

3-7 融合 异构分组融合：


K_1
q_{1i}
a_1
Q组1
每个组有一个注意力数值，
保证相加后是归一化的
q_{1i}
a_2
K_2
q_{2i}
Q组2
。 。 。 。 。 。 。 。 。 。 。
进一步，q可以相加或者拼接
q_{2i}

4 应用任务

- 模型普适性
  - 序列剪枝 + 序列分类任务，标注，分割：Length-adaptive

- 优化方法设计

优化方法设计

- 无需训练的压缩
  - 量化
    - **brecq框架，采用一个block的输出误差求解类中心和模型的其他参数。**
  - 稀疏
    - 直接稀疏注意力矩阵，不微调，对于在大数据集上预训练的模型很适用，现有资源很难在大数据集上训练

- 优化目标设计

优化目标设计

- **1 压缩算法指导-依赖目标**
- **2 蒸馏结合：数据高效**
- 3 避免过平滑：避免token越来越相似
  - Refiner DiverseVIT

- 优化目标普适性
  - 适用于密集预测任务：Evovit LVVIT
- 给定压缩率的设置
  - 每个压缩模块单独的压缩率：Dvit Adavit
  - 整体的压缩率：DGE

1 压缩算法指导-依赖目标

- 最小损失函数：引入额外参数，最小化loss优化这些参数
- 最大熵-信息量：不依赖参数，无需设置可导的gate变量
- 最小重构误差：不依赖参数，无需设置可导的gate变量

2 结合蒸馏

- 数据高效：
  - DearKD-CVPR’22
  - TinyKD-ECCV’22 Bootstrap-ECCV’22 TinyVIT-ECCV’22

- 优化变量设计

优化变量设置

- 参数化模块vs非参数化模块：需要额外训练开销，数据隐私保护问题。
  - 1 参数化模块：
    - Gate变量必须和某个输出相乘，才能求导；因此Gate变量无法减少训练开销；
    - 1-1 阈值：阈值作为参数，可用于分配每层的压缩率
      - LTP-KDD'22
    - 1-2 重要性得分:一个**得分**计算模块，依赖参数，需要设置额外的可导的gate变量
      - 依赖输入
        - 当前gate对应的特征： TCFormer DGE PAUMER-BMVC'22
        - 所有gate对应的特征 ： Dvit SPVit FBS DMI EBERT DGNET TR-BERT IA-RED SE-NET Real2BIN; TokenLearner
        - 不依赖特征 Funnel SVITE Slimming Sinkhorn SparseQKV
    - 1-3 位置:一个**位置**计算模块，依赖参数，需要设置额外的可导的gate变量
      - 输入压缩：SP-Vit DAT-CVPR'22
      - 输出压缩：OVA DCN Deformable-DETR-ICLR'21 Repgraph-ECCV'20
    - 1-4 聚类权重：SIT 见输入压缩-软注意力
  - 2 非参数化模块：无需学习，直接测试
    - EViT TP TOME-ICLR'23 EDPT-NIPS'22