

论 文 汇 报

Language Knowledge-Assisted Representation Learning for Skeleton-Based Action Recognition

(基于骨架的动作识别中的语言知识辅助表示学习)

► 汇报时间：2024.04.20

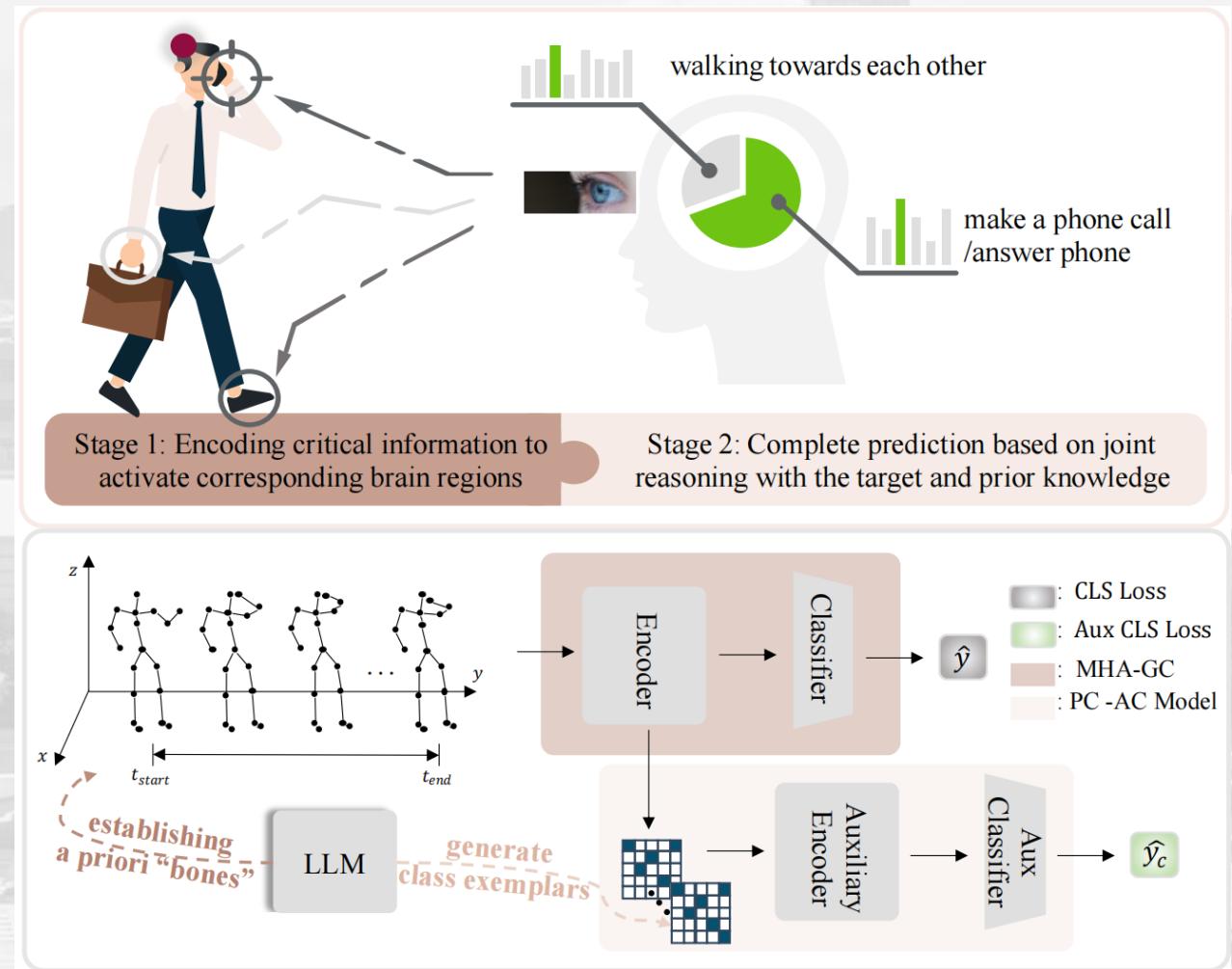
► 汇报人：梁琨

目 录

CONTENT

壹	<u>行为识别任务综述</u>	03
貳	<u>大语言模型BERT</u>	12
參	<u>模型结构</u>	16
肆	<u>实验设计及结果</u>	20

整体结构+突出创新点



1 Part One

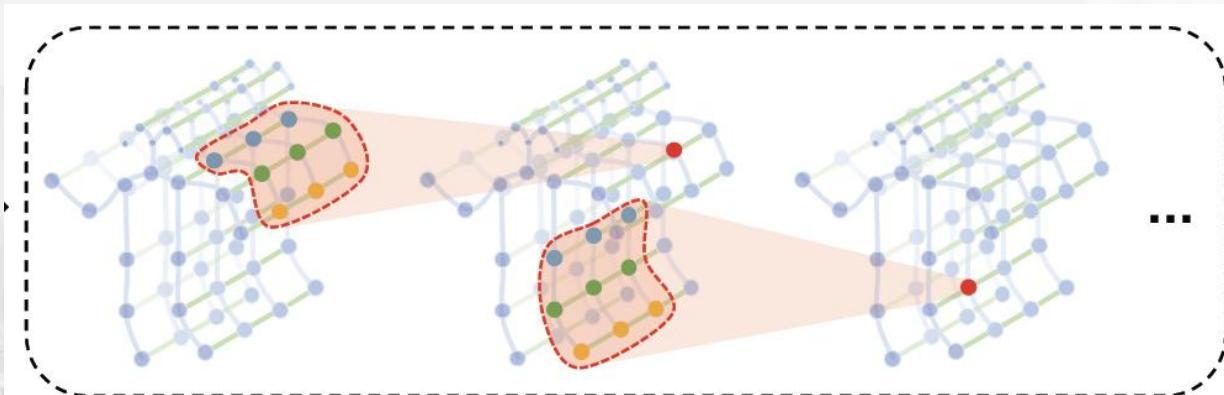
行为识别任务综述

介绍本篇论文中行为识别的方案——时空图卷积网络，分别考虑时间和空间维度建模，最终串行连接，形成模型的主体部分。

骨骼数据结构

行为识别中骨骼数据的提取和使用

XDU



骨骼数据结构的优点：

数据量小，隐私性高，蕴含信息丰富

骨骼数据结构的缺点：

不如图像视频数据，可以直接使用，骨骼提取仍需要模型来完成，如openpose

常用的数据格式：

N

C

T

V

M

批次大小

骨骼点特征数

动作维持帧数

关节点个数

图片中人物（骨架）个数

二维：x, y

三维：x, y, z

(以图片帧为单位的批次)

骨架数据的多种表示方案

XDU

人体骨架数据的多维度表示

人体骨架的
多维度表示

关节：原始关节坐标

骨骼：通过对具有物理连接的关节坐标进行差分得到的向量

关节运动：关节在时间维度上的差分

骨骼运动：骨骼在时间维度上的差分

对于关节 i , 坐标可以表示为 (x_i, y_i, z_i)

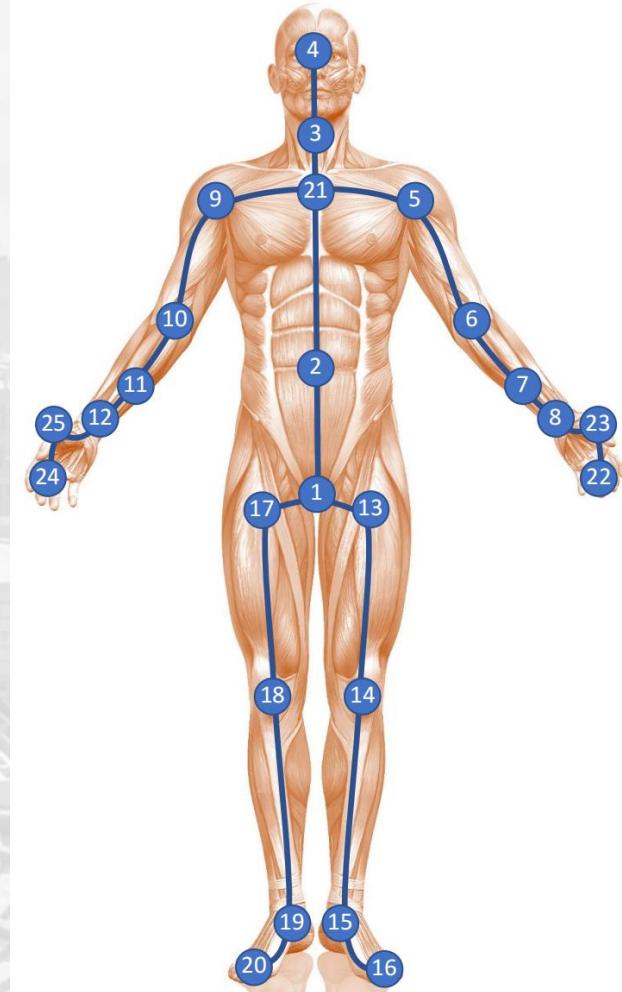
对于骨骼，其可以看做是连接两个相邻的关节之间的通道，因此，对于关节 i, j , 骨骼可以表示为

$$(x_i - x_j, y_i - y_j, z_i - z_j)$$

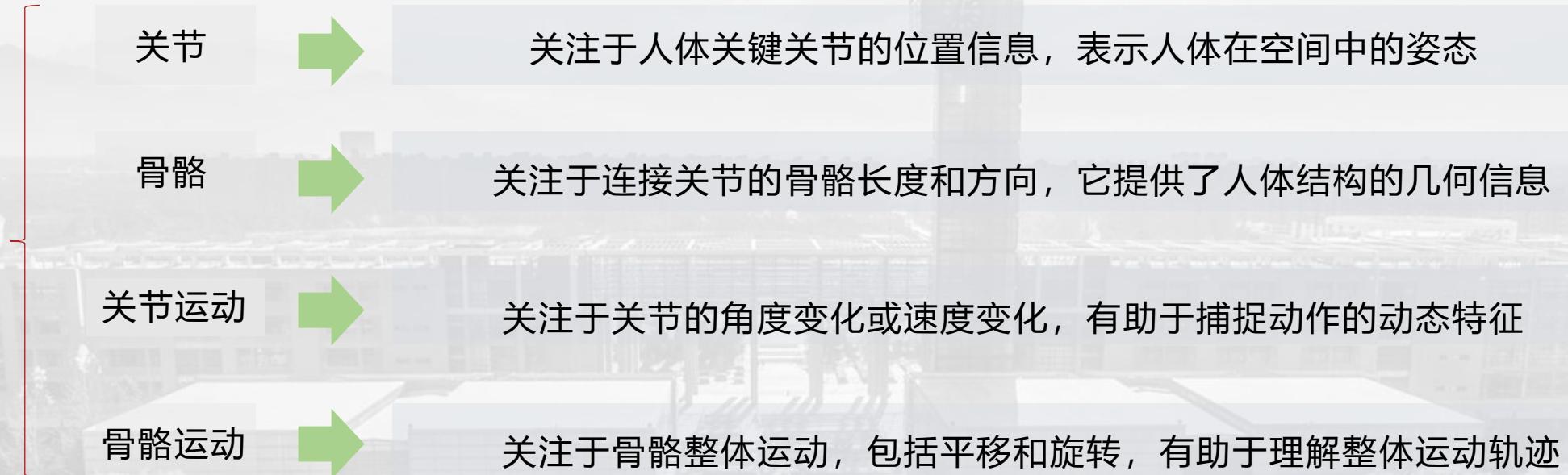
矩阵形式表示为 $\tilde{X}_t = (I - B) X_t$

其中， X 为 t 时刻的关节特征， \tilde{X}_t 为 t 时刻的骨骼特征

B 为骨矩阵即关节的邻接矩阵



NTU RGB+D的骨架结构



特征层融合
决策层融合

多流集成，融合模型基于多种骨架的推理结果

实验方案：

多种骨架数据**分别利用LA-GCN模型**进行训练，推理时对每个结果进行**加权平均**

多流集成，融合模型基于多种骨架的推理结果

实验方案：

多种骨架数据分别利用LA-GCN模型进行训练，推理时对每个结果进行加权平均

```
dataset = arg.dataset
if 'UCLA' in arg.dataset:
    arg.alpha = [2.5, 1.5, 3.5, 1, 0.5, 1]
    label = []
    with open('./data/' + 'NW-UCLA/' + 'val_label.pkl', 'rb') as f:
        data_info = pickle.load(f)
        for index in range(len(data_info)):
            info = data_info[index]
            label.append(int(info['label']) - 1)
elif 'ntu120' in arg.dataset:
    arg.alpha = [3, 2, 1, 0.5, 2, 1.5]
    if 'xsub' in arg.dataset:
        npz_data = np.load('./data/' + 'ntu120/' + 'NTU120_CSub.npz')
        label = np.where(npz_data['y_test'] > 0)[1]
    elif 'xset' in arg.dataset:
        npz_data = np.load('./data/' + 'ntu120/' + 'NTU120_CSet.npz')
        label = np.where(npz_data['y_test'] > 0)[1]
elif 'ntu' in arg.dataset:
    arg.alpha = [2, 2.5, 2, 0.5, 1.5, 1.5]
    if 'xsub' in arg.dataset:
        npz_data = np.load('./data/' + 'ntu/' + 'NTU60_CS.npz')
        label = np.where(npz_data['y_test'] > 0)[1]
    elif 'xview' in arg.dataset:
        npz_data = np.load('./data/' + 'ntu/' + 'NTU60_CV.npz')
        label = np.where(npz_data['y_test'] > 0)[1]
else:
    raise NotImplementedError
```

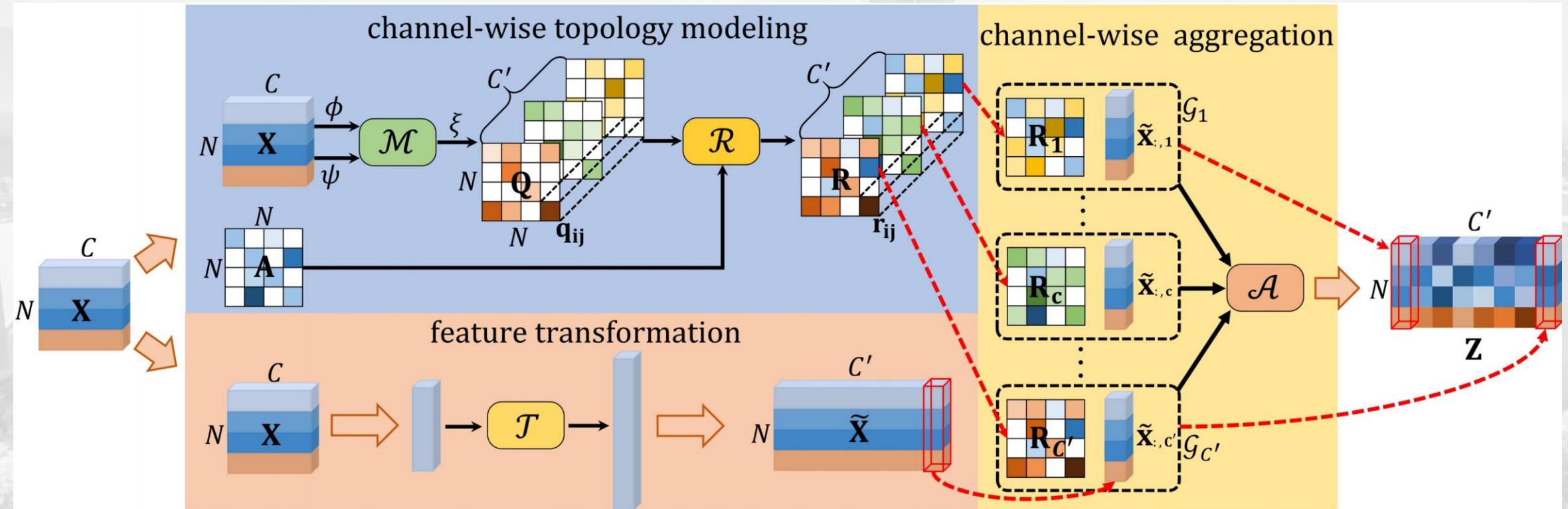
```
for i in tqdm(range(len(label)), disable=args.slient):
    l = label[i]
    _, r11 = r1[i]
    _, r22 = r2[i]
    _, r33 = r3[i]
    _, r44 = r4[i]
    _, r55 = r5[i]
    _, r66 = r6[i]
    r = r11 * arg.alpha[0] + r22 * arg.alpha[1] + r33 * arg.alpha[2] \
        + r44 * arg.alpha[3] + r55 * arg.alpha[4] + r66 * arg.alpha[5]
    rank_5 = r.argsort()[-5:]
    right_num_5 += int(int(l) in rank_5)
    r = np.argmax(r)
    right_num += int(r == int(l))
    total_num += 1
acc = right_num / total_num
acc5 = right_num_5 / total_num

print('Top1 Acc: {:.4f}%'.format(acc * 100))
print('Top5 Acc: {:.4f}%'.format(acc5 * 100))
```

4+2

针对于“骨骼”这类骨架数据，利用LLM
的先验知识构建了一种新的骨架表示

Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition



Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition[3]

Static Topology-shared GCs

Static Topology-non-shared GCs

Dynamic Topology-shared GCs

Dynamic Topology-non-shared GCs

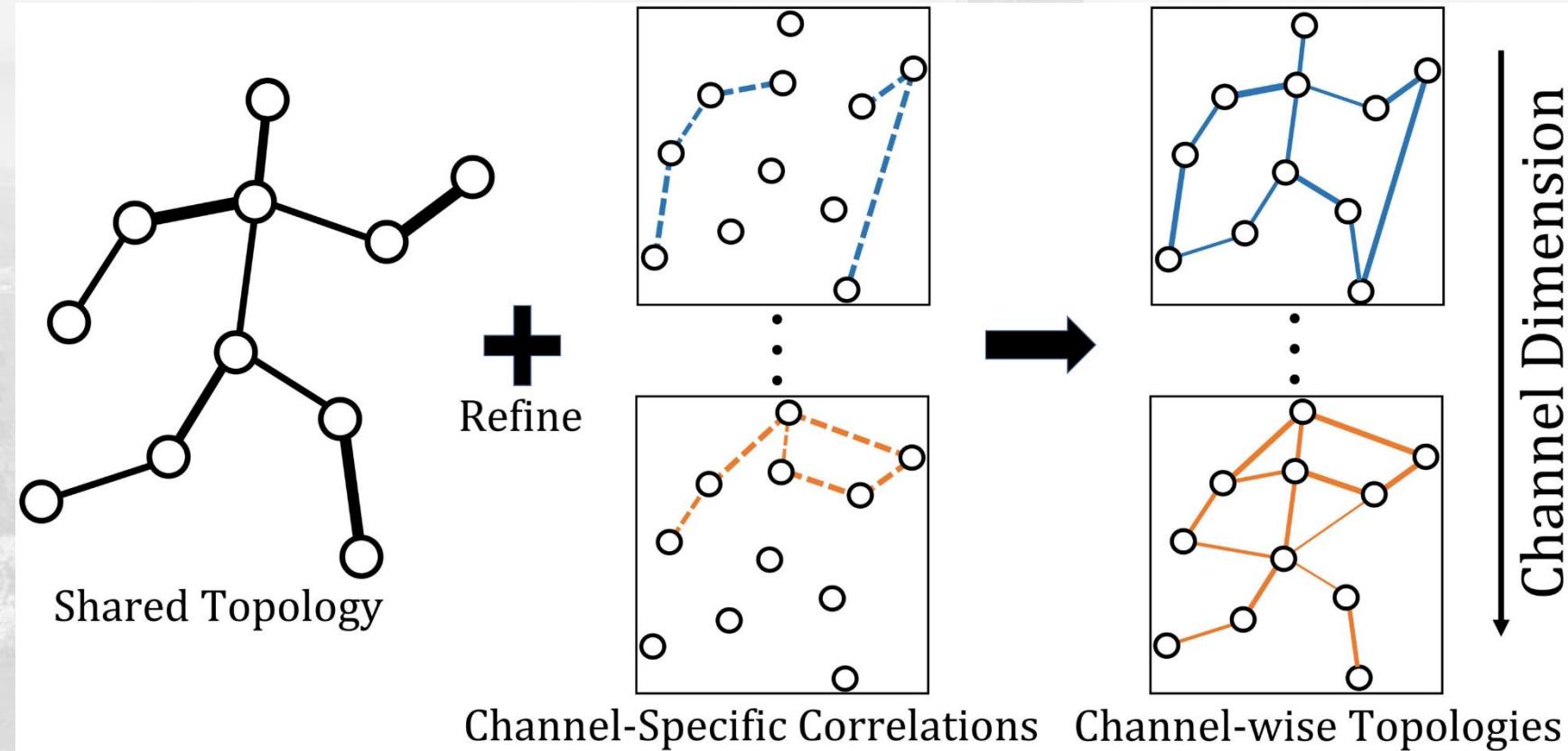
GCNs已成功应用于基于骨架的动作识别中，其中大部分遵循论文[4]的特征更新规则，**特征更新规则**包括两个步骤：

- (1) Transform features into high-level representations;
- (2) Aggregate features according to graph topology.

根据拓扑结构的差异，基于GCN的方法可以分为如下：

- (1) 根据推理过程中拓扑结构是否被动态调整，基于GCN的方法可以分为静态方法和动态方法；
- (2) 根据拓扑是否在不同的通道上共享，基于GCN的方法可以分为拓扑共享方法和拓扑非共享方法。

LA-GCN属于动态非共享拓扑



动态非共享拓扑：共享拓扑+特定通道的关系矩阵 == 通道级拓扑

2 Part Two

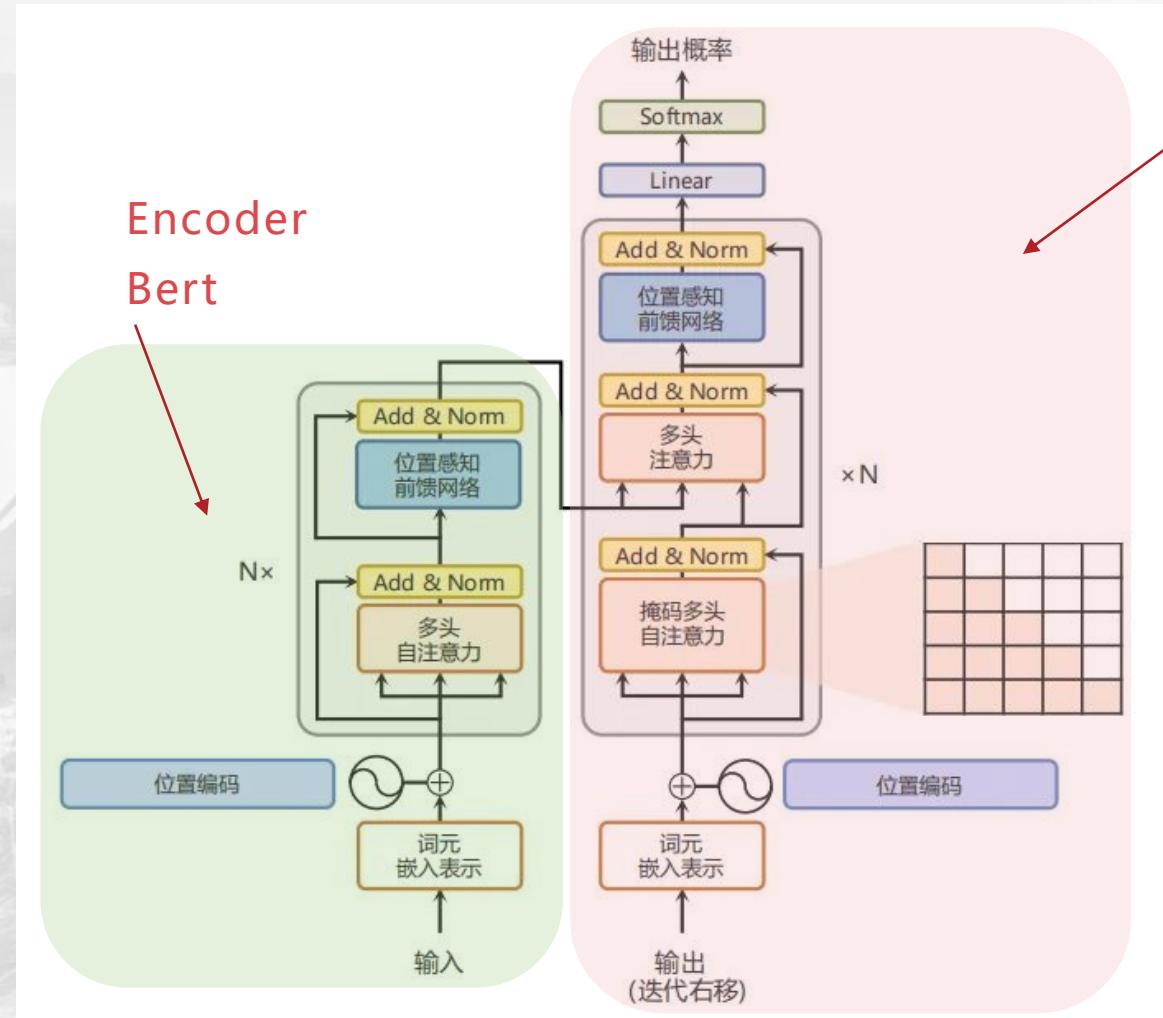
大语言模型BERT

简要介绍论文中涉及到的大模型理论知识

Transformer综述

用于NLP任务，特别是Sequence-to-Sequence的学习问题

XDU

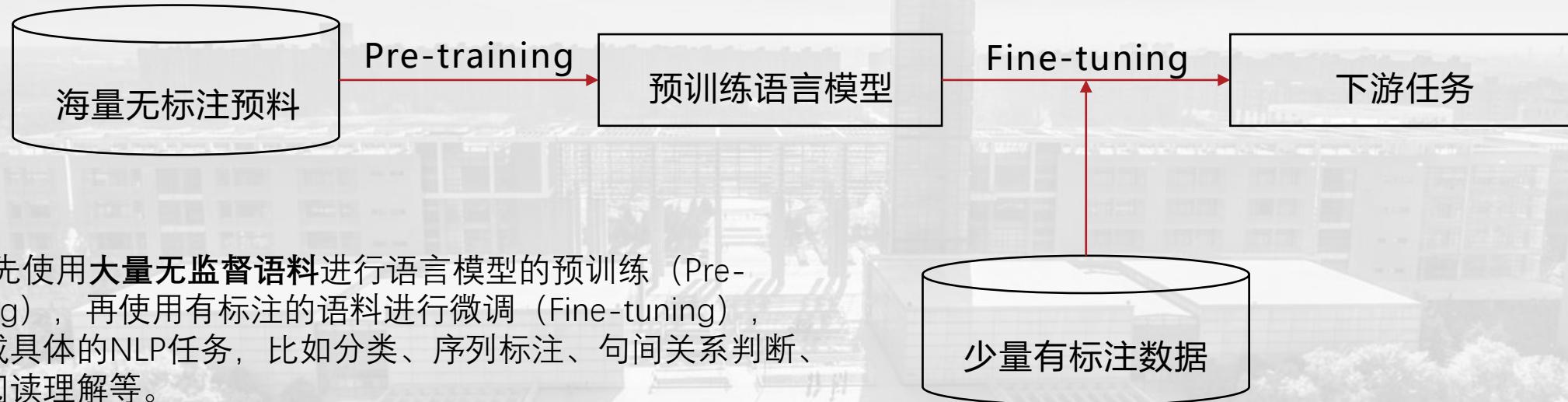


Decoder
GPT

Transformer的核心构成包括：

- 自注意力机制 (Self-Attention Mechanism)
- 多头注意力 (Multi-Head Attention)
- 位置编码 (Positional Encoding)
- 编码器-解码器架构 (Encoder-Decoder Architecture)
- 残差连接 (Residual Connections)
- 层归一化 (Layer Normalization)

预训练模型：



定义：

首先使用大量无监督语料进行语言模型的预训练（Pre-training），再使用有标注的语料进行微调（Fine-tuning），来完成具体的NLP任务，比如分类、序列标注、句间关系判断、机器阅读理解等。

优势：

- 近乎无限量的数据；
- 无需人工标注；
- 一次学习，多次复用；
- 学习到的表征可以在多个任务中进行快速迁移。

自监督方式
预训练方式：
• 基于特征的 (Feature-based)
• 基于微调的 (Fine-tuning)

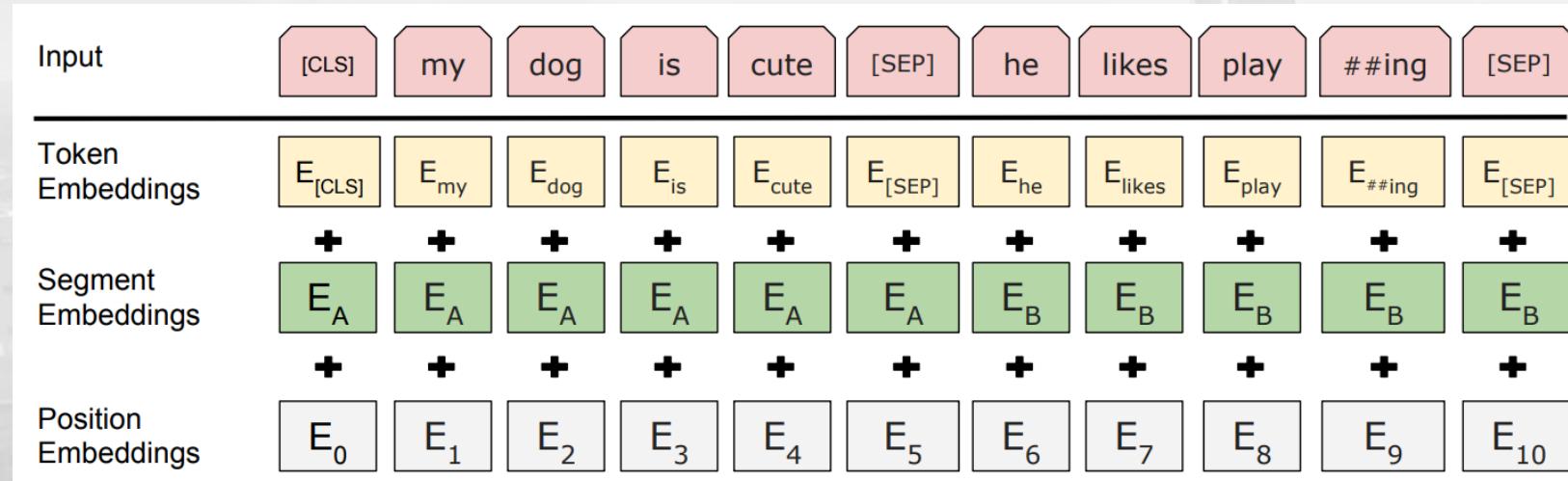


分类任务
回归任务

有监督学习 VS 自监督学习

MLM
NSP

输入向量



将单词转换为固定维度的向量

区分句子对的上下句

标记句中每个词的位置

Segment Embeddings 示例：

[CLS] 我 的 狗 很 可 爱 [SEP] 企 鹅 不 擅 长 飞 行 [SEP]
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

任务一：MLM (Masked Language Modeling)

MASK 策略示例：

对于语句“my dog is hairy”，随机把句中15%的token替换为以下内容：

80%的几率被替换成[MASK]：

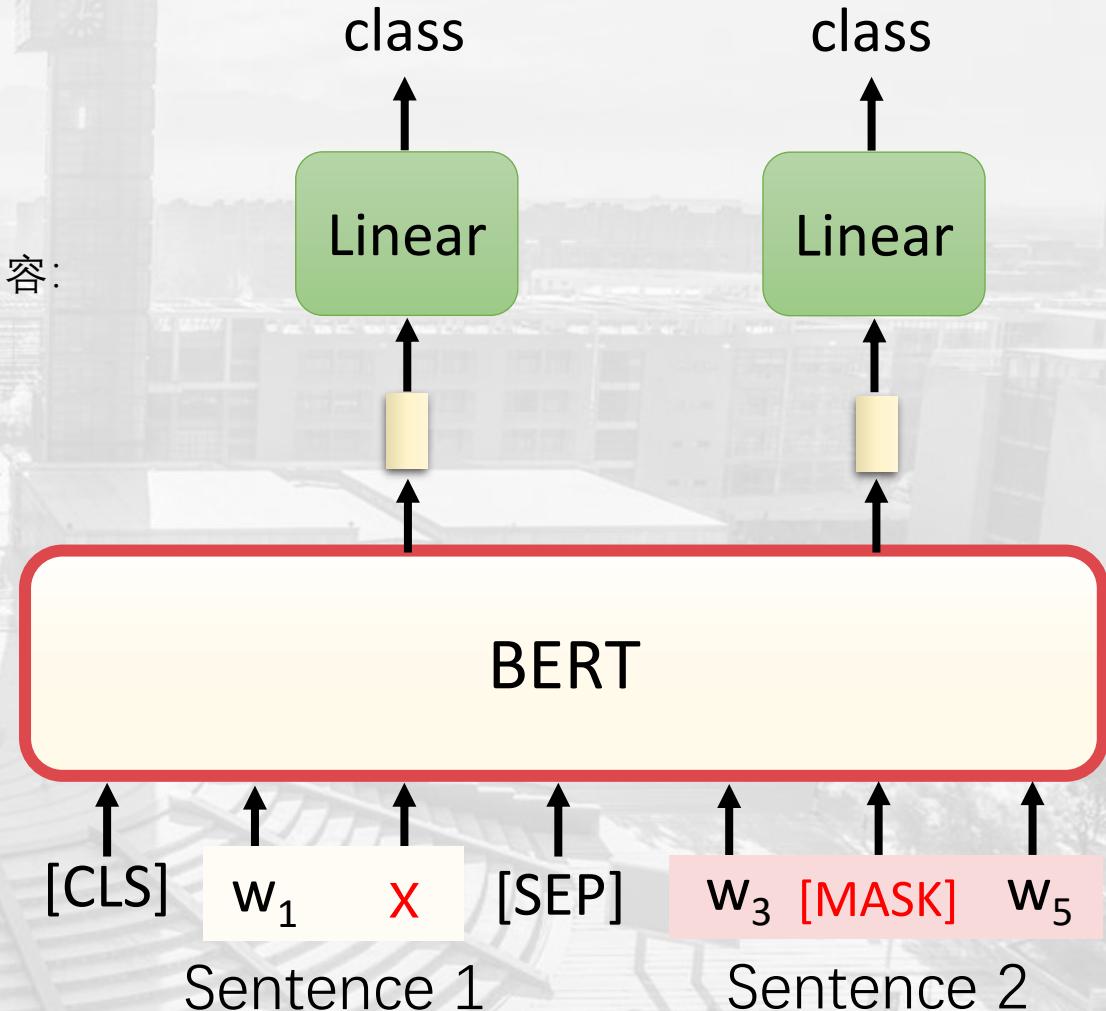
“my dog is hairy” → “my dog is [MASK]”

10%的几率被替换成其他token：

“my dog is hairy” → “my dog is apple”

10%的几率原封不动：

“my dog is hairy” → “my dog is hairy”



Bidirectional Encoder Representations from Transformers.

任务二：NSP (Next Sentence Prediction)

正负句子对样本：

50% 的正样本：训练语料库中的两个连续段落

50% 的负样本：来自不同文档的两个随机段落

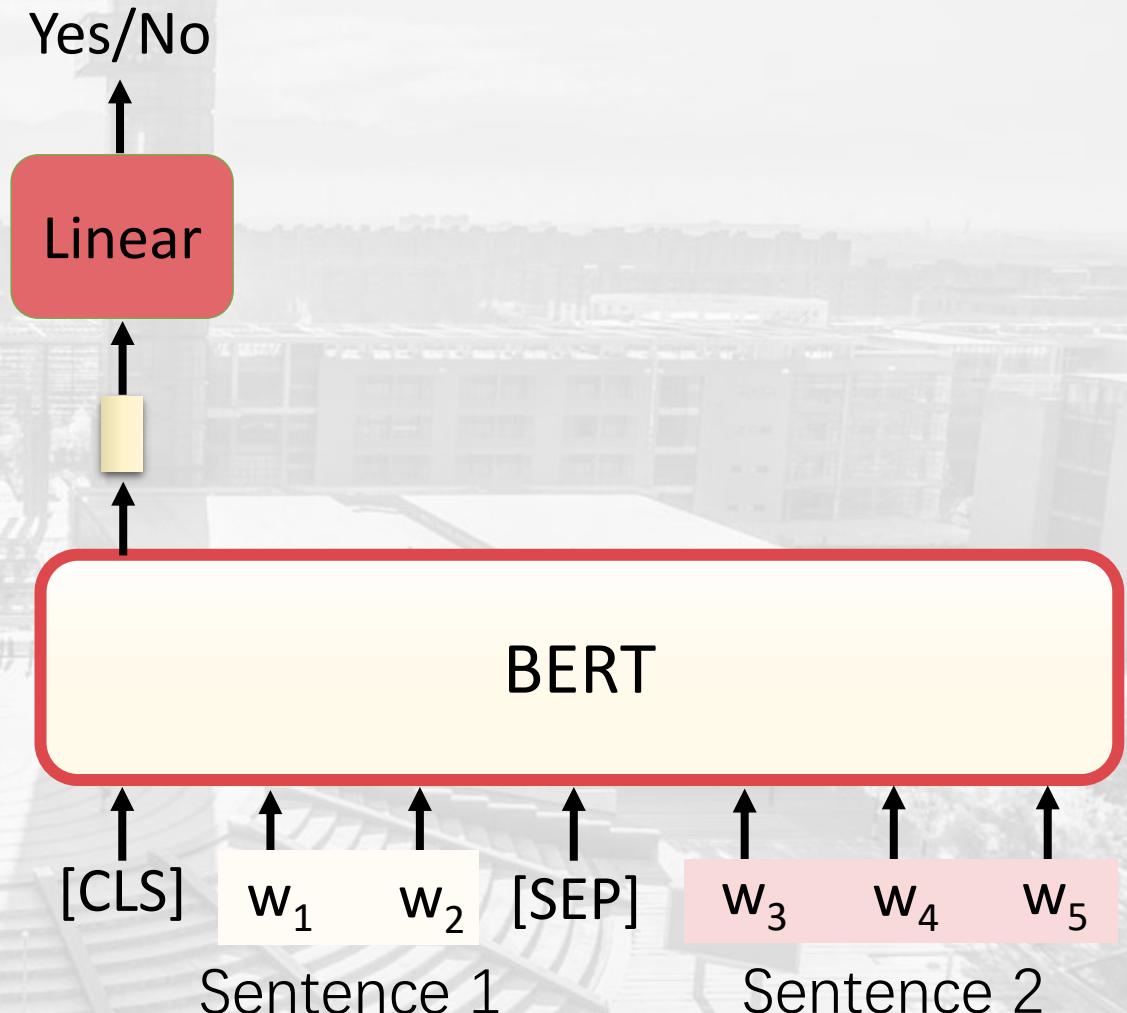
示例：

Input: [CLS] 博学而笃志 [SEP] 切问而近思 [SEP]

Target: Yes

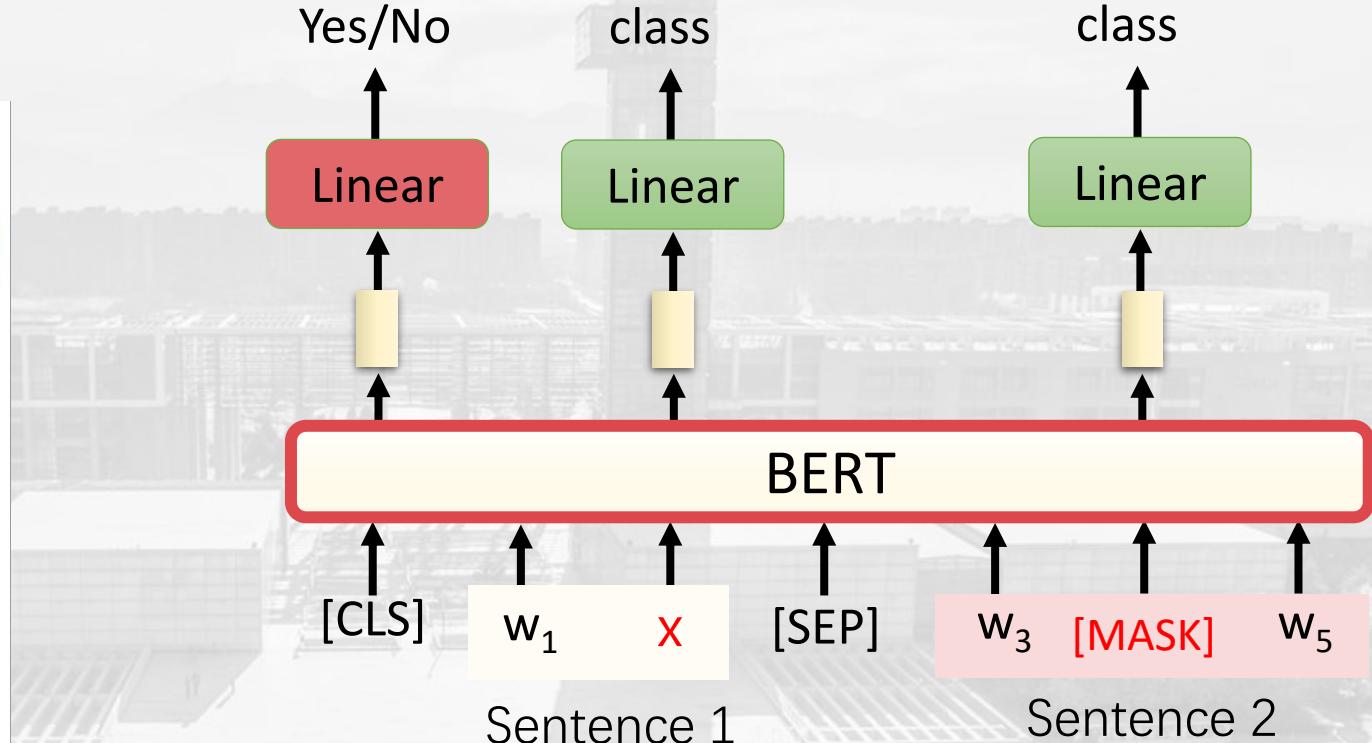
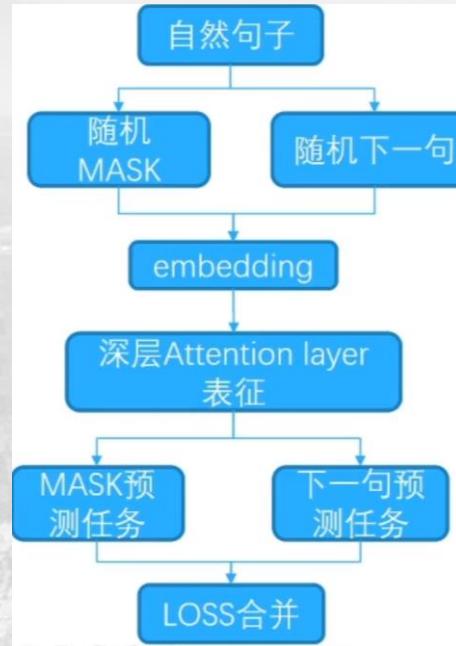
Input: [CLS] 博学而笃志 [SEP] 今天风好大[SEP]

Target: No



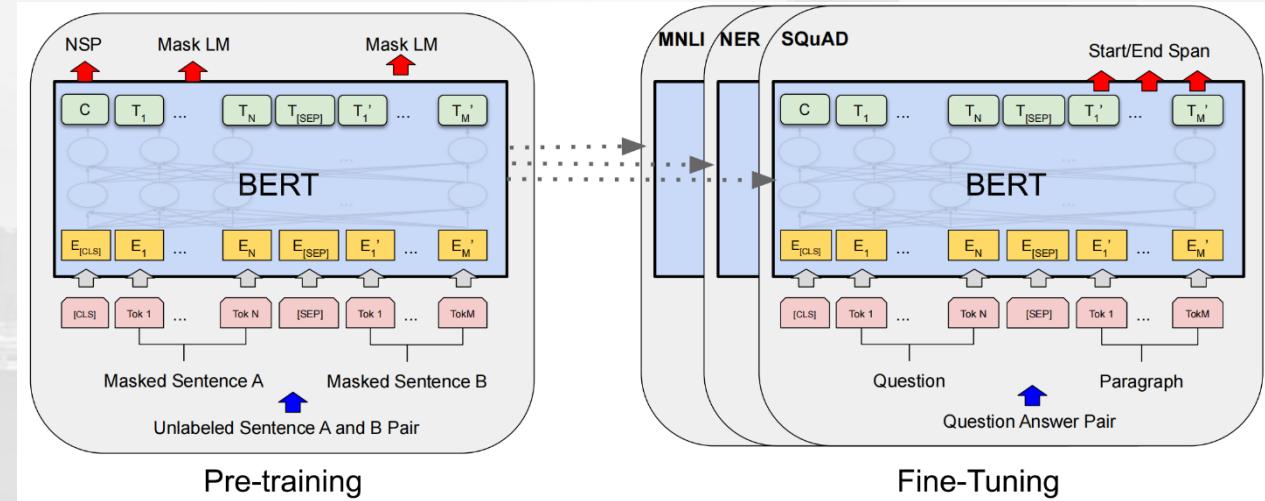
Bidirectional Encoder Representations from Transformers.

Multi-Task Learning



- Input: [CLS] calculus is a branch of math [SEP] panda is native to [MASK] central china [SEP]
- Targets: **false**, south
- -----
- Input: [CLS] calculus is a [MASK] of math [SEP] it [MASK] developed by newton and leibniz [SEP]
- Targets: **true**, branch, was

Bidirectional Encoder Representations from Transformers.



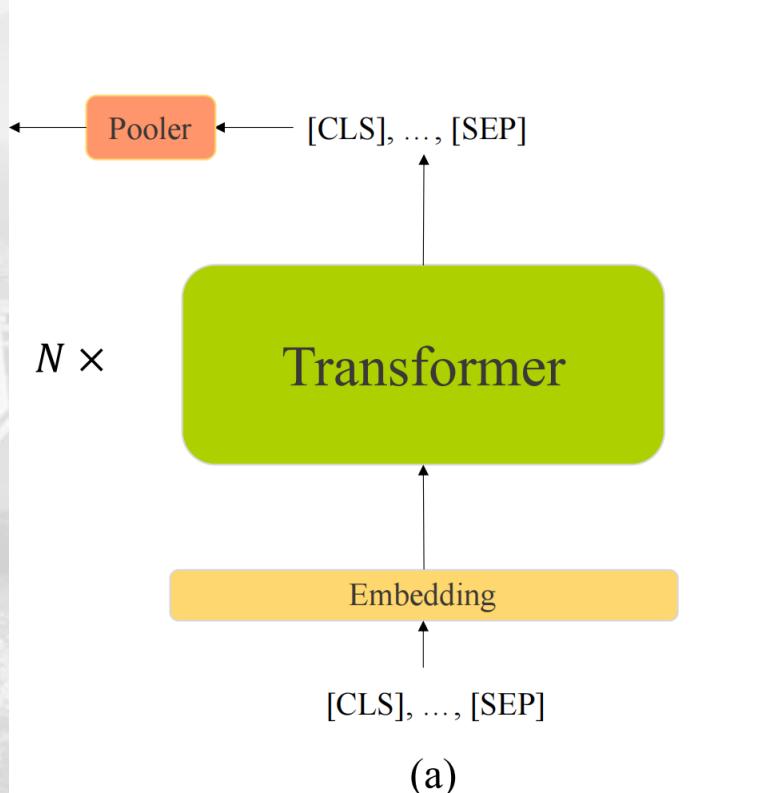
上图为原文中给出的预训练和微调示意图，和GPT一样，BERT也采用两阶段式训练方法：

- 第一阶段：使用易获取的大规模无标签语料来训练语言模型（LM），为了能够同时利于 token-level task 和 sentence-level task，作者采用如下两种预训练任务来建立语言模型（多任务训练目标），从而更好的支持下游任务：
 - 掩码语言模型：Masked Language Model (MLM)，目的是提高模型的语义理解能力；
 - 下句预测：Next Sentence Prediction (NSP)，目的是训练句子之间的理解能力；
- 第二阶段：利用下游任务的有标签训练语料，进行微调训练。

A Global Prior Relation Graph Generated by LLM

GPR+CPR

XDU



使用BERT来提取每一类动作标签和所有关节的文本特征。在预训练过程中BERT的损失值包括填空和输入两句话来预测后者是否是前一句的句子。模型的最后一层的输出用于完成填空，Pooler部分的输出用于下一个句子的预测。**由于Pooler之后的特征包含了整个句子的语义，因此它们可以在执行一般的文本分类等任务时直接使用。在本文中，也选择了Pooler后的特征作为我们的骨架节点的特征。**

A Global Prior Relation Graph Generated by LLM

GPR+CPR

XDU

给定M个动作类别和N个人类的关节点，对每个节点提取特征。

输入：包含**动作类别 + 骨骼点名称**

例如：

p1: [J] function in [C].

p2: What happens to [J] when a person is [C]?

p3: What will [J] act like when [C]?

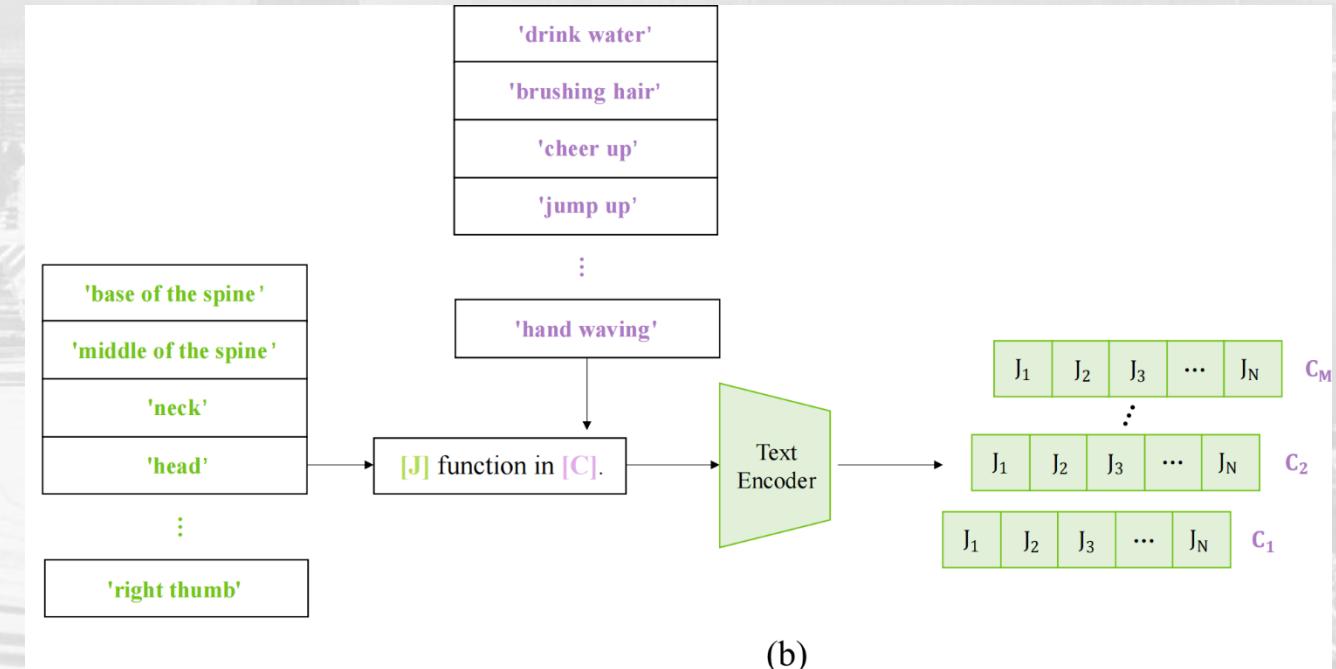
p4: When [C][J] of human body.

p5: When [C] what will [J] act like?

p6: When a person is [C], [J] is in motion.

对于动作 C_i ，LLM的文本编码器得到相应的输出特征

$$C_i \in \mathbb{R}^{N \times C} (i=1, 2, \dots, M)$$



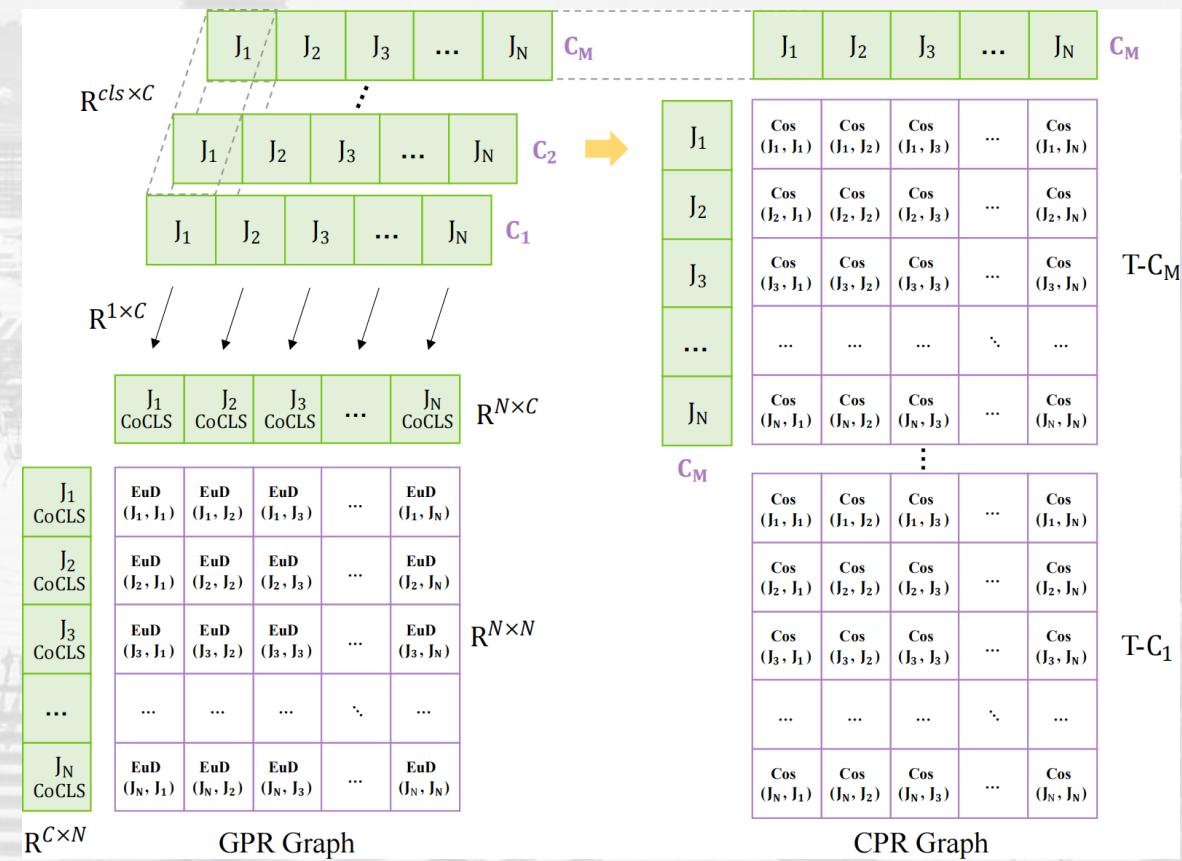
CPR的计算方法

XDU

A class relationship topology graph containing a priori node relationships

构造由文本特征生成的类别拓扑图。对于动作 $C_i (i = 1, 2, \dots, M)$ ，
设节点 $k (k = 1, 2, \dots, N)$ 的文本特征为 $J_k \in \mathbb{R}^{1 \times C}$
将两两文本特征逐个组合起来计算余弦相似度

对于每个动作 C_i ，都可以生成唯一一个 $T - C_i$ 拓扑图与之对应，该图主要包含了任意两个关节点关于动作 C_i 的相似和相关程度。
我们将 $T - C_i$ 称为一个类拓扑范例，并假设它们包含应该存在于动作识别操作中的节点关系



GPR的计算方法

XDU

A global inter-node relationship topology graph GPR-Graph

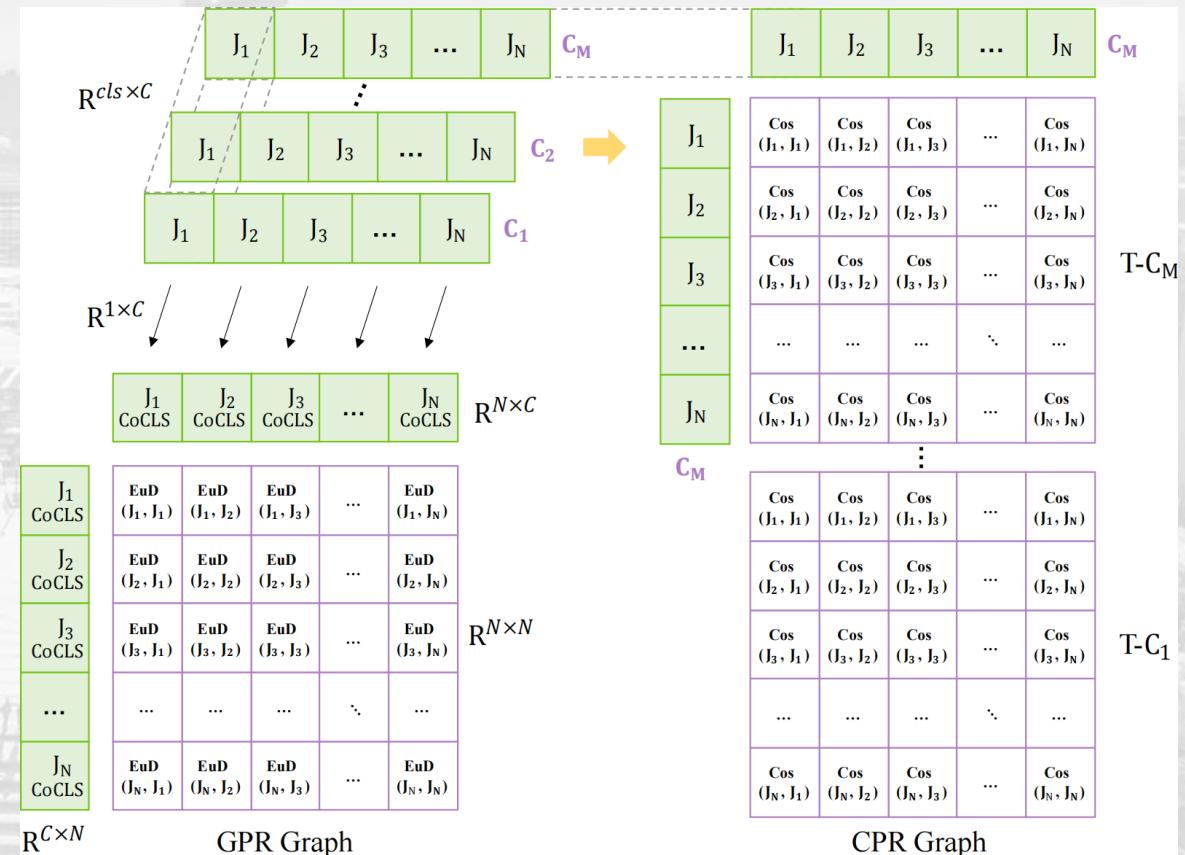
GPR包含LLM中语义知识的全局节点间的关系，通过GPR-Graph来指导骨架表示的生成

对于节点 $J_k (k = 1, 2, \dots, N)$, 将其在 M 个动作类别上的文本特征进行平均，得到该节点的类中心向量：

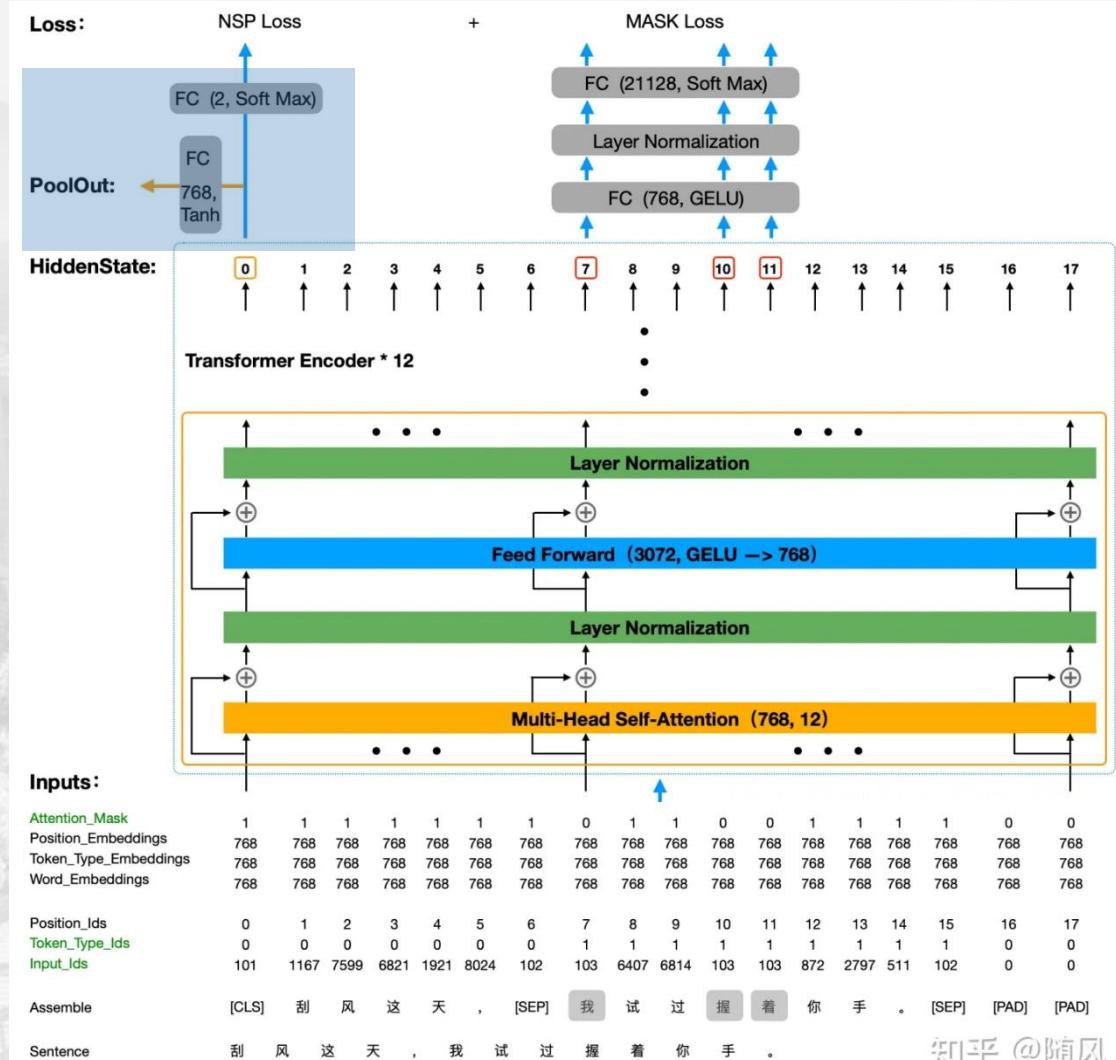
$$J_k^{CoCLS} = \frac{1}{M} \sum_{j=1}^M J_k^{C_j}$$

$$J^{CoCLS} \in \mathbb{R}^{N \times C}$$

计算任意两个节点的类中心向量之间的相关性，计算方法采取欧氏距离



Bidirectional Encoder Representations from Transformers.



为什么说大语言模型LLM可以辅助行为识别

我们将节点之间的差异视为附加表示的“骨”数据。例如，NTU数据集包含25个节点，如果我们将任何两个节点之间的差向量视为“骨”，就有 $C_{25}^2 = 300$ 个“骨骼”。

对于任意两个关节点 i, j ，在时刻 t ，设关节特征分别为 $J_i = (x_i, y_i, z_i)$, $J_j = (x_j, y_j, z_j)$ ，设骨骼特征为 $M_t = (x_t, y_t, z_t)$ 其中满足

$$M = J_i - J_j = (x_i - x_j, y_i - y_j, z_i - z_j)$$

定义任意关节 i, j 之间的骨骼标准差：

骨骼的各个特征在时间维度上的标准差的均值，即 $b_{std} = mean(\sigma(X_t), \sigma(Y_t), \sigma(Z_t))$

经过计算，物理骨骼矩阵中的骨骼链接具有较小的标准偏差总和。基于此，选择标准差之和中最小的骨组合作为新的“骨”表示。**使用GPR图的信息**，即节点间距离对骨骼矢量进行加权，并提取**std 求和最小的骨骼**作为新的骨骼表示。

A Global Prior Relation Graph Generated by LLM

GPR+CPR

XDU

生成先验拓扑图的方法：

GPR图：

通过计算关节的类中心特征相关性；得到全局先验信息，对应于每个节点的动作类中心之间的距离

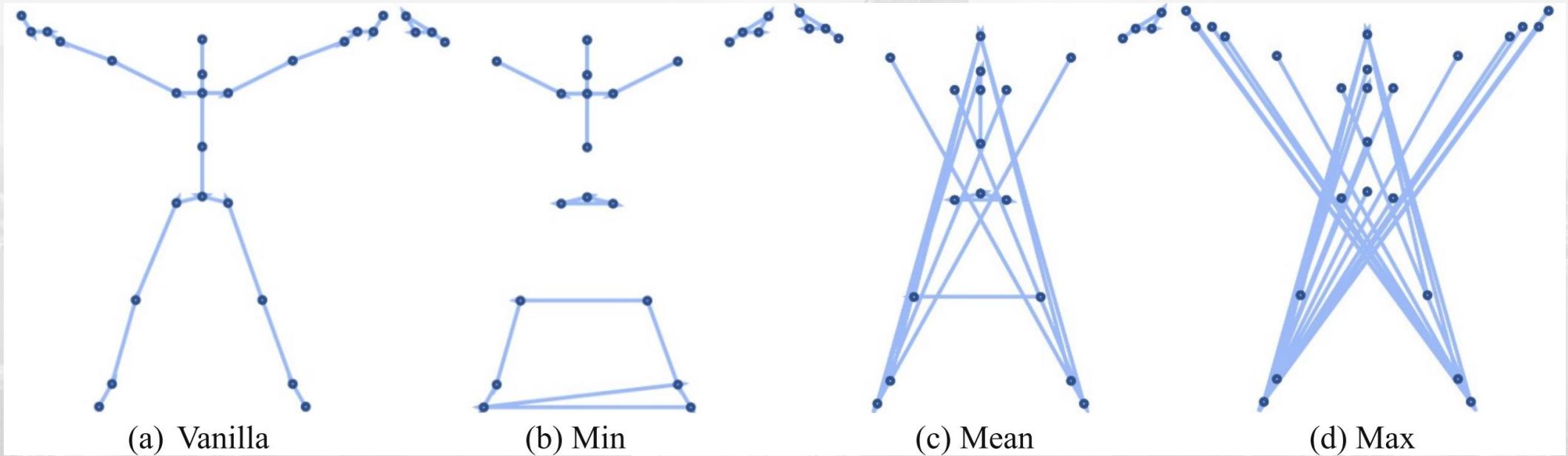
CPR图：

计算每个动作的节点特征之间的相关性

LLM → BERT → 文本特征 → 特征相关性 →

GPR：先验骨架模态表示
对输入骨架序列进行加权表示

CPR：先验一致性辅助分类
PC-AC模块
额外的分支协同监督



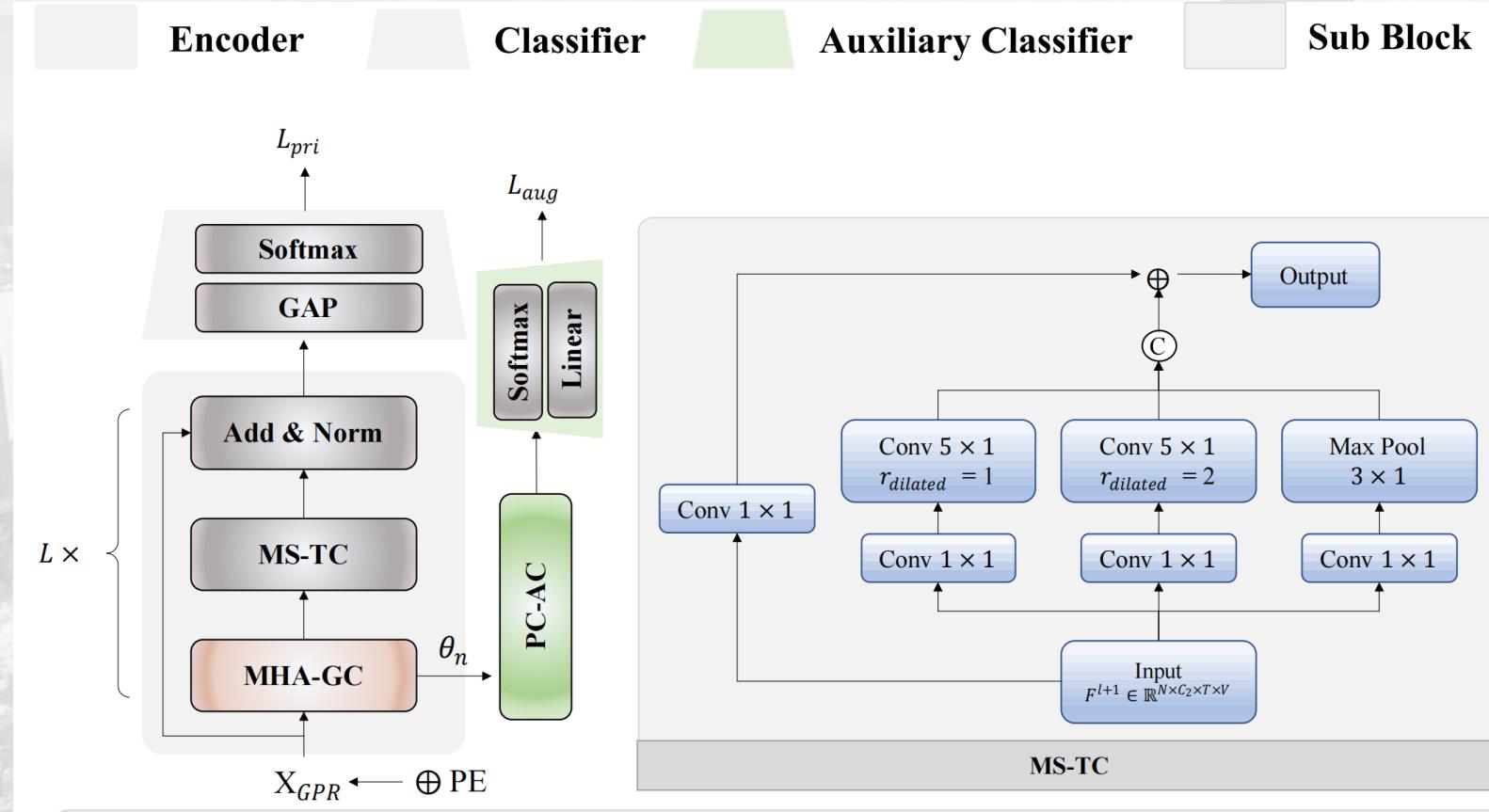
先验模态表示法的两个主要优点：

- (1) 它可以捕捉没有直接连接的远程节点之间的交互。这种交互可能有助于识别某些动作。
- (2) 它包含类属性的上下文依赖性

3 *Part Three*

模型结构

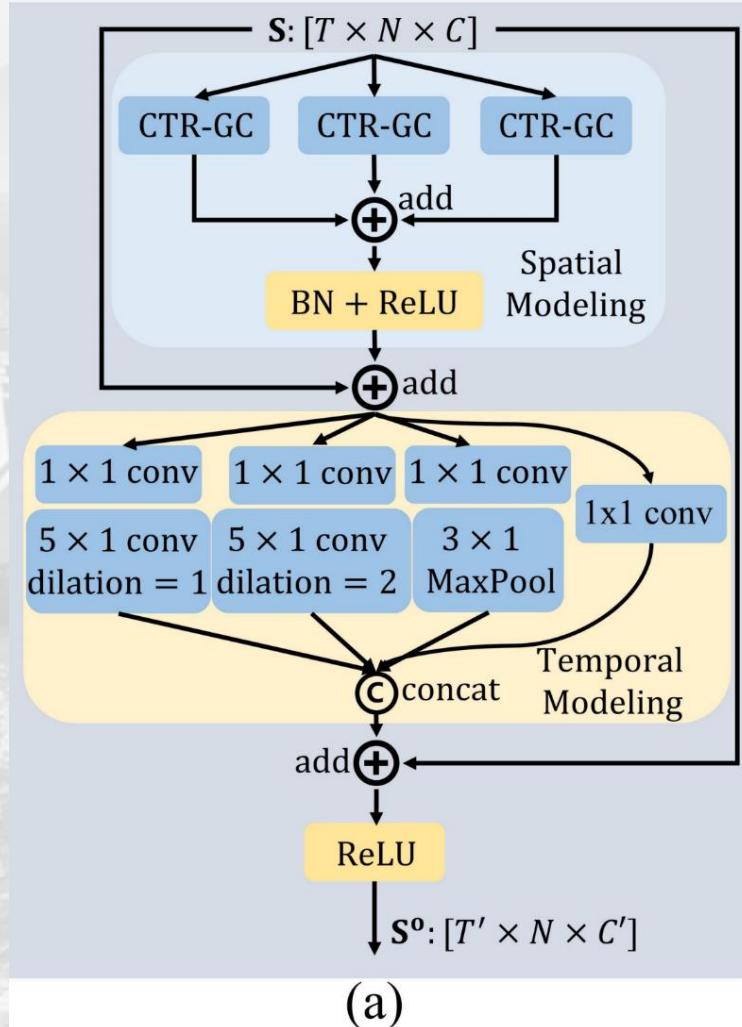
介绍论文中的模型结构，并补充两个创新点，以及引用论文与其进行比较分析。

**Temporal Conv:**

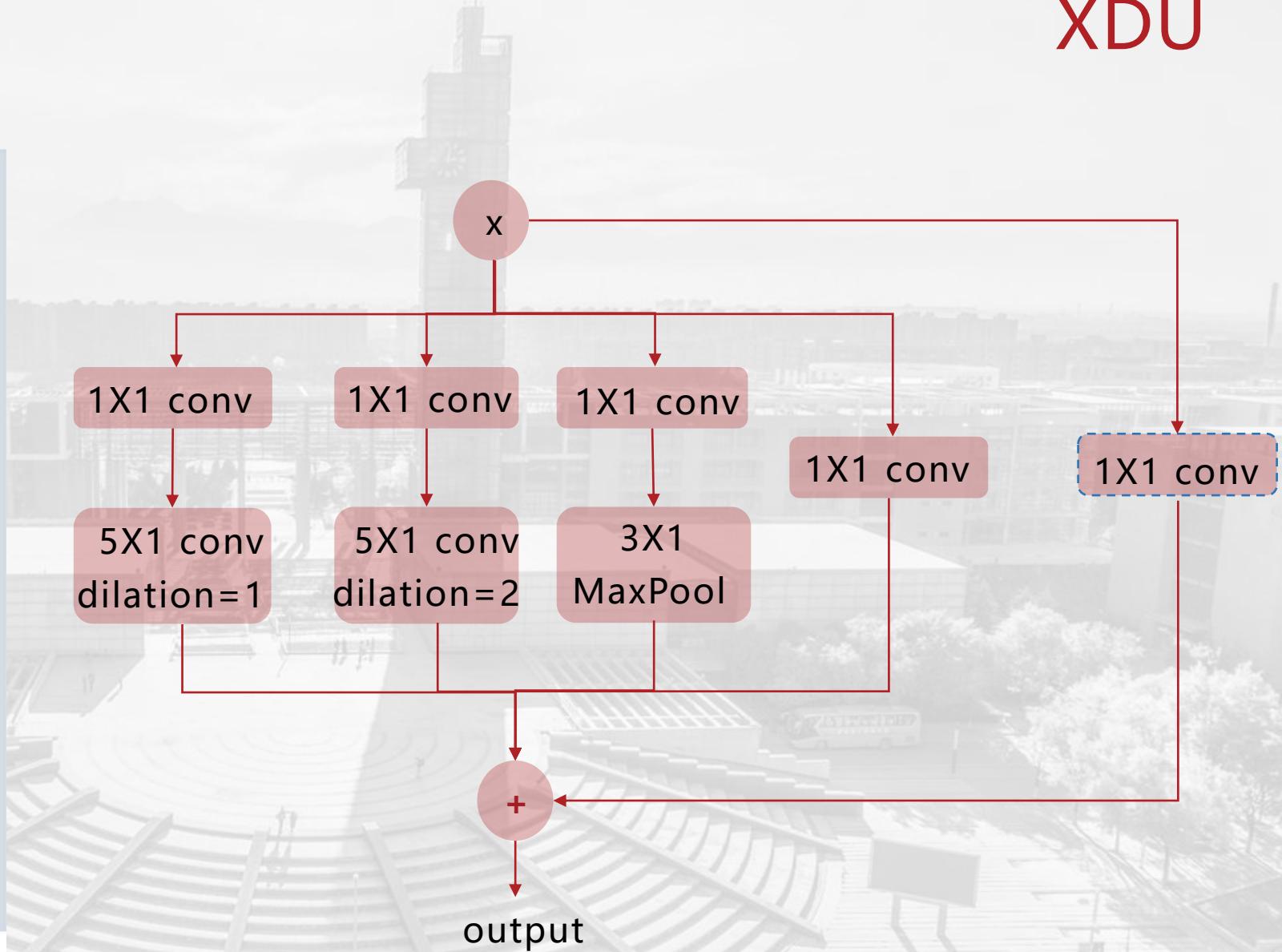
- ✓ 引入多尺度的时间卷积模块，分别用来提取不同时间帧长度的动作特征
- ✓ 提取出的不同长度的特征进行concatenate操作
- ✓ 引入residual残差连接，避免模型深度的增加造成梯度消失，同时有助于原始特征的传播

模型结构

MS-TC 时间维度



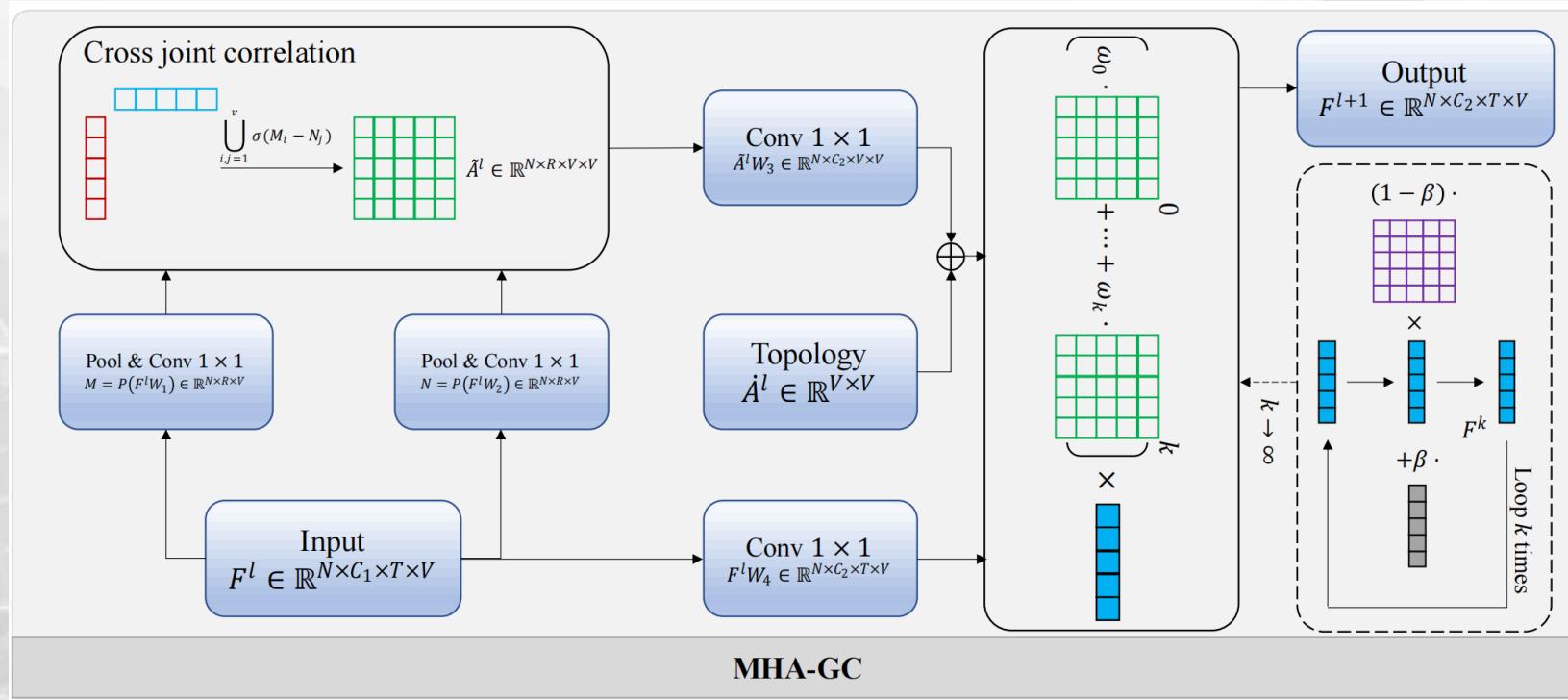
(a)



模型结构

MHA-GC 空间维度

XDU



Spacial Conv:

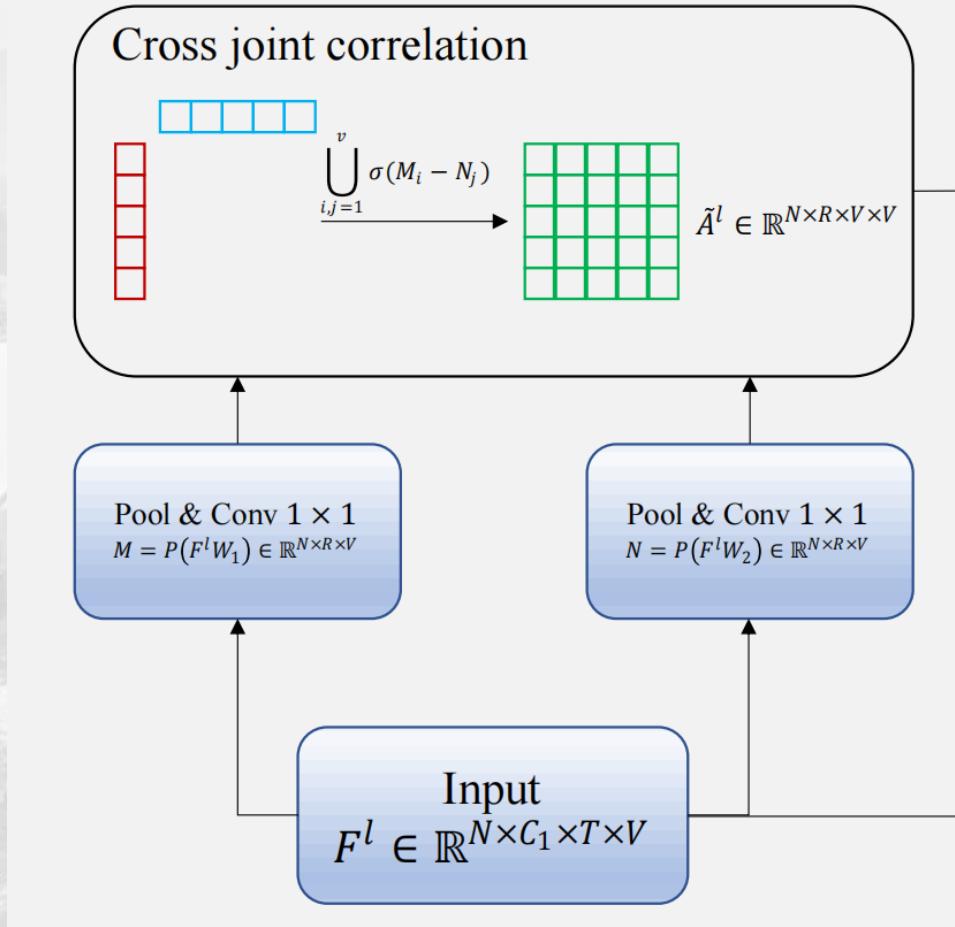
$$\text{Input } \left\{ \begin{array}{l} F^0 = X_{GPR} W_0 + PE \\ F^{(0)} \in \mathbb{R}^{N \times C \times T \times V} \\ PE \in \mathbb{R}^{C \times V} \end{array} \right.$$

经过Conv和Pool:

Channel-Specific Correlations ← Cross joint correlation ←

$$\left[\begin{array}{l} M = P(F^l W_1) \in \mathbb{R}^{N \times R \times V} \\ N = P(F^l W_2) \in \mathbb{R}^{N \times R \times V} \end{array} \right]$$

Cross joint correlation



Feature vectors:

$$M, N \in \mathbb{R}^{N \times R \times V}$$

Any pair of vertices (v_i, v_j) in M and N is computed separately to obtain the first-order neighborhood information

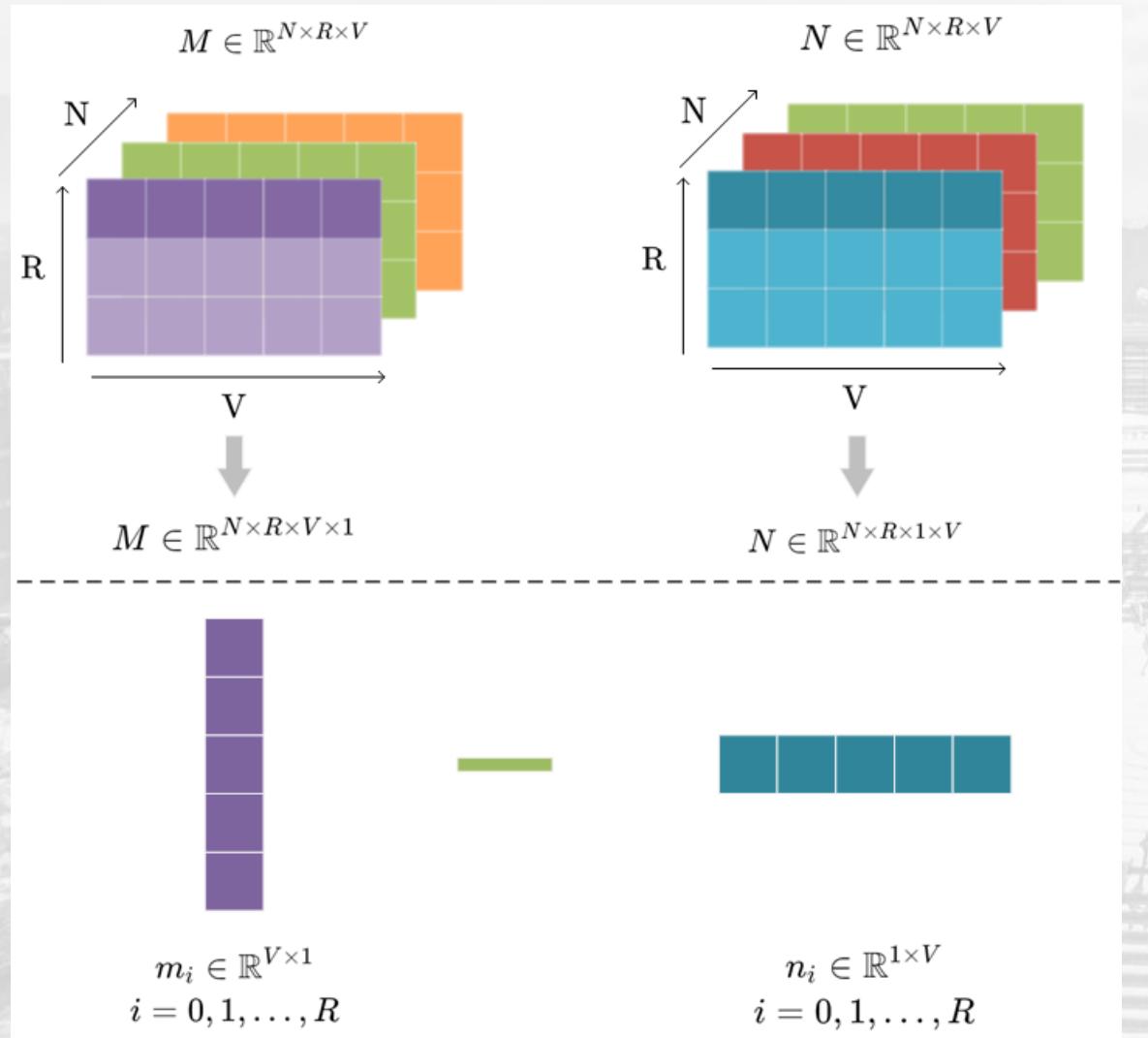
$$\tilde{A}^l = \sum_{i,j=1}^v \sigma(M_i - N_j)$$

where σ is the activation function, v indicates all nodes, and \tilde{A}_{ij}^l denotes the messages aggregation from node j to node i .

模型结构

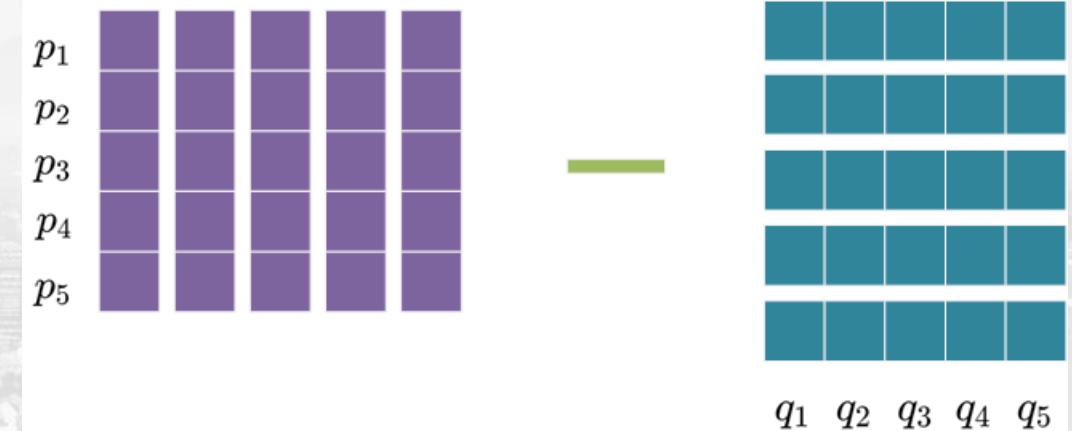
XDU

Cross joint correlation

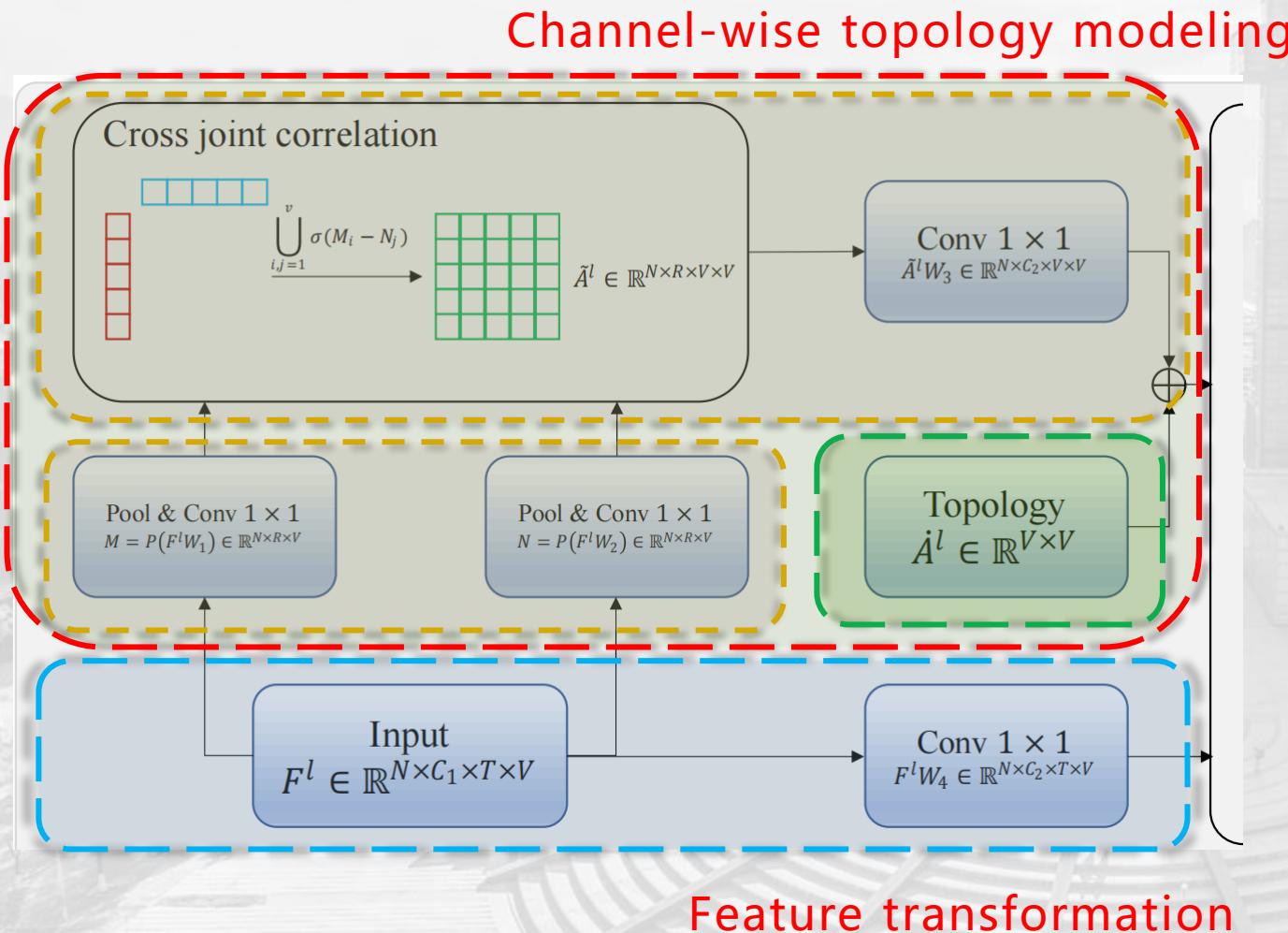


西安电子科技大学

Tensor Broadcast



$$Z = \begin{pmatrix} p_1 - q_1 & p_1 - q_2 & p_1 - q_3 & p_1 - q_4 & p_1 - q_5 \\ p_2 - q_1 & p_2 - q_2 & p_2 - q_3 & p_2 - q_4 & p_2 - q_5 \\ p_3 - q_1 & p_3 - q_2 & p_3 - q_3 & p_3 - q_4 & p_3 - q_5 \\ p_4 - q_1 & p_4 - q_2 & p_4 - q_3 & p_4 - q_4 & p_4 - q_5 \\ p_5 - q_1 & p_5 - q_2 & p_5 - q_3 & p_5 - q_4 & p_5 - q_5 \end{pmatrix}$$



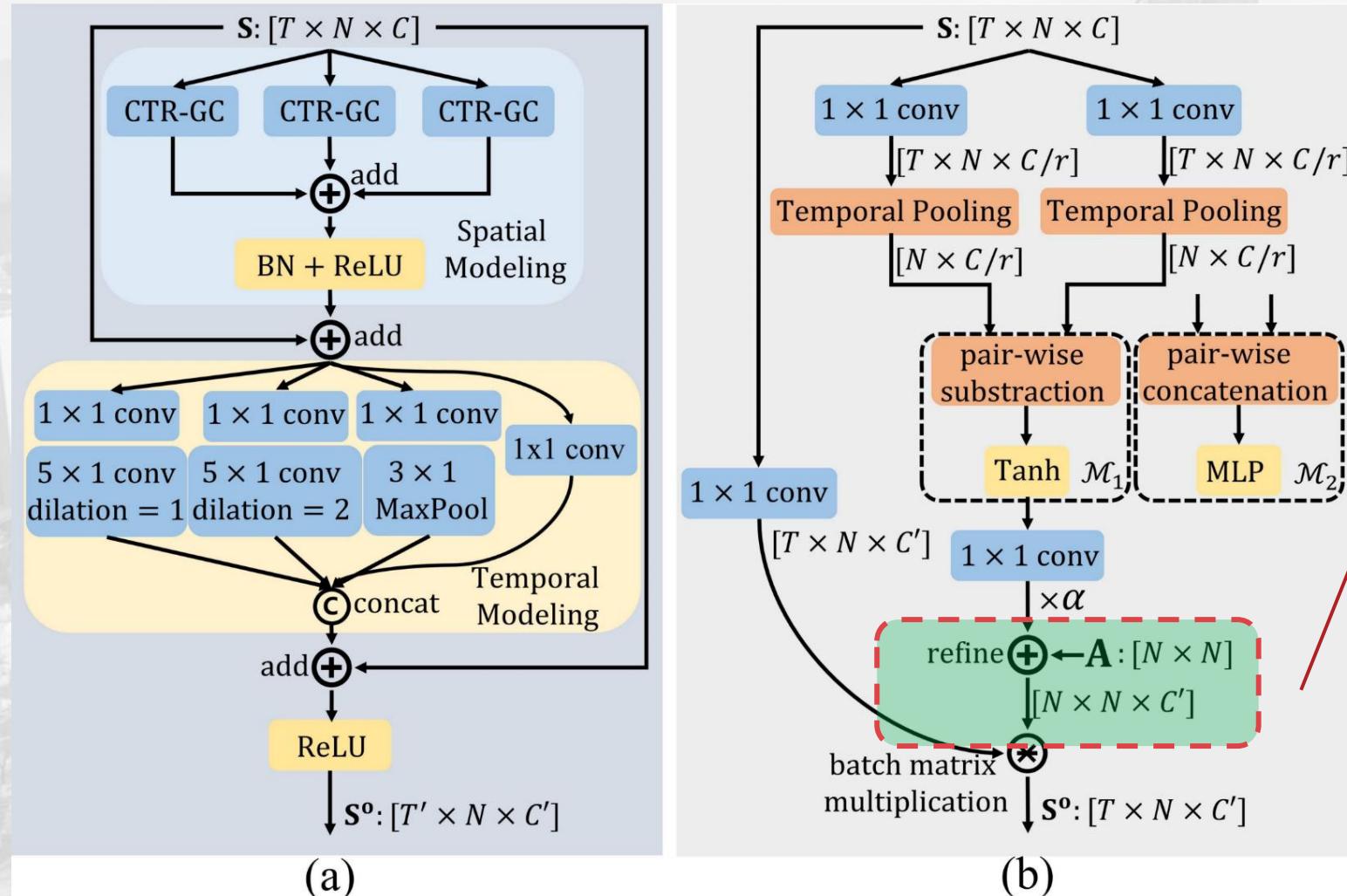
Spacial Conv:

$$\bar{A}^l = \dot{A}^l + \gamma \tilde{A}^l W_3$$

Shared Topology

Channel-Specific Correlations

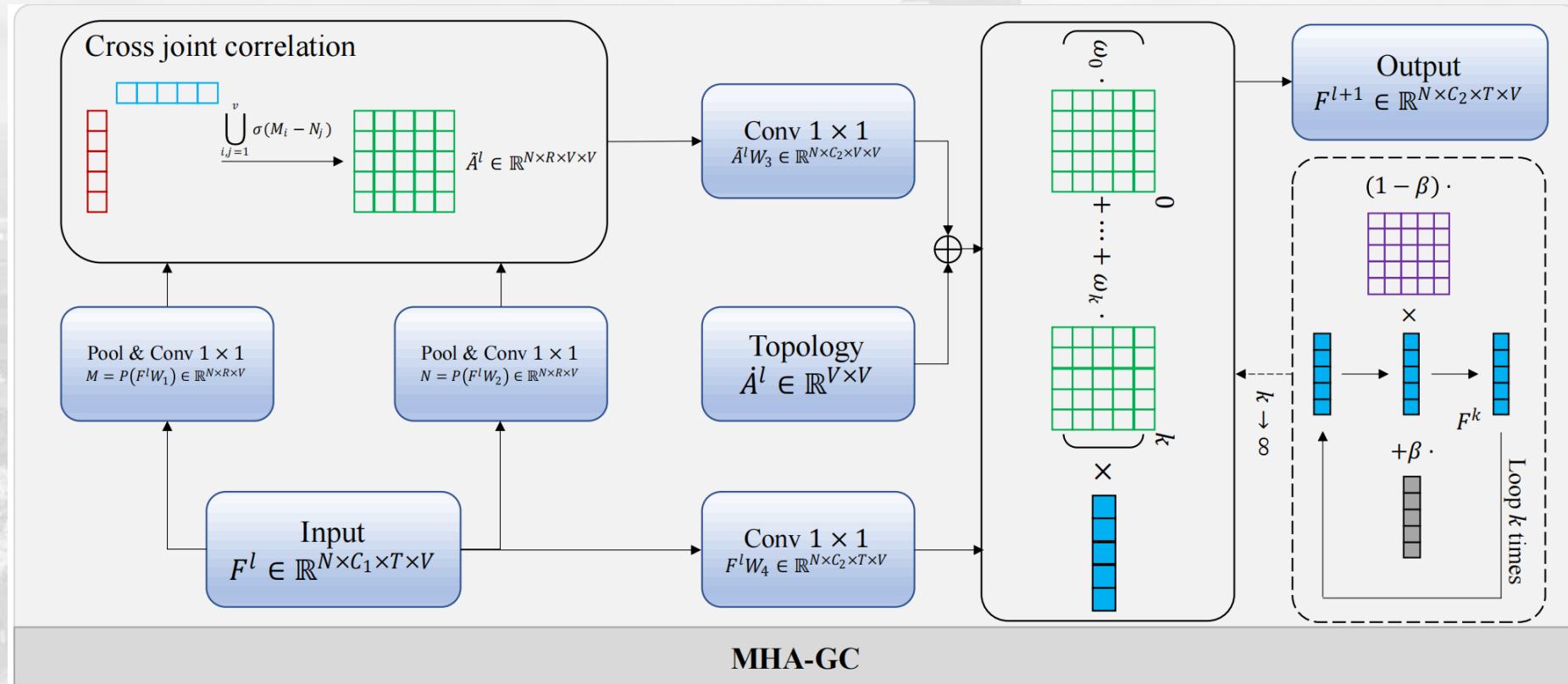
Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition



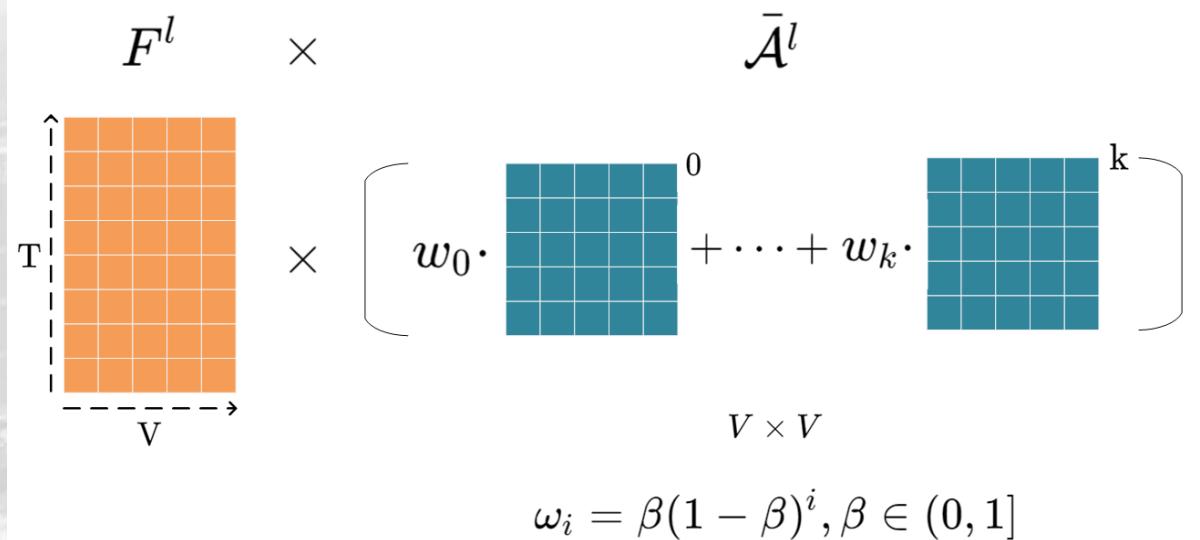
$$\bar{A}^l = \dot{A}^l + \gamma \tilde{A}^l W_3$$

Before batch matrix
Multiplication and
feature aggregation:

K-Loop



$$F^l \in \mathbb{R}^{N \times C_2 \times T \times V}$$



$$F^{l+1} \in \mathbb{R}^{N \times C_2 \times T \times V}$$

$$\left\{ \begin{array}{l} \bar{\mathcal{A}}^l = \sum_{i=0}^k \omega_i \bar{A}^i \\ \omega_i = \beta(1 - \beta)^i, \beta \in (0, 1] \\ \omega_i \quad \bar{A}^i \text{ 的延迟因子} \\ F^{l+1} = \sigma(\bar{\mathcal{A}}^l F^l W_4^l) \end{array} \right.$$

(在特征聚合前, F^l 已经完成输出层的线性映射, 即 $F^l = F^l W_4^l$)



$$0*1 + 2*1 + 1*0 + 0*1 = 2$$

$0*1$ 代表 $v1-v1$: 0条通路 $v1-v3$: 1条通路，分步相乘 0条通路

$2*1$ 代表 $v1-v2$: 2条通路 $v2-v3$: 1条通路，分步相乘 2条通路

$1*0$ 代表 $v1-v3$: 1条通路 $v3-v3$: 0条通路，分步相乘 0条通路

$0*1$ 代表 $v1-v4$: 0条通路 $v4-v3$: 1条通路，分步相乘 0条通路

最后分类相加就是最后的 $v1-v3$ 长度为 2 的通路数。

建立原本关系弱或无关系的节点之间的联系

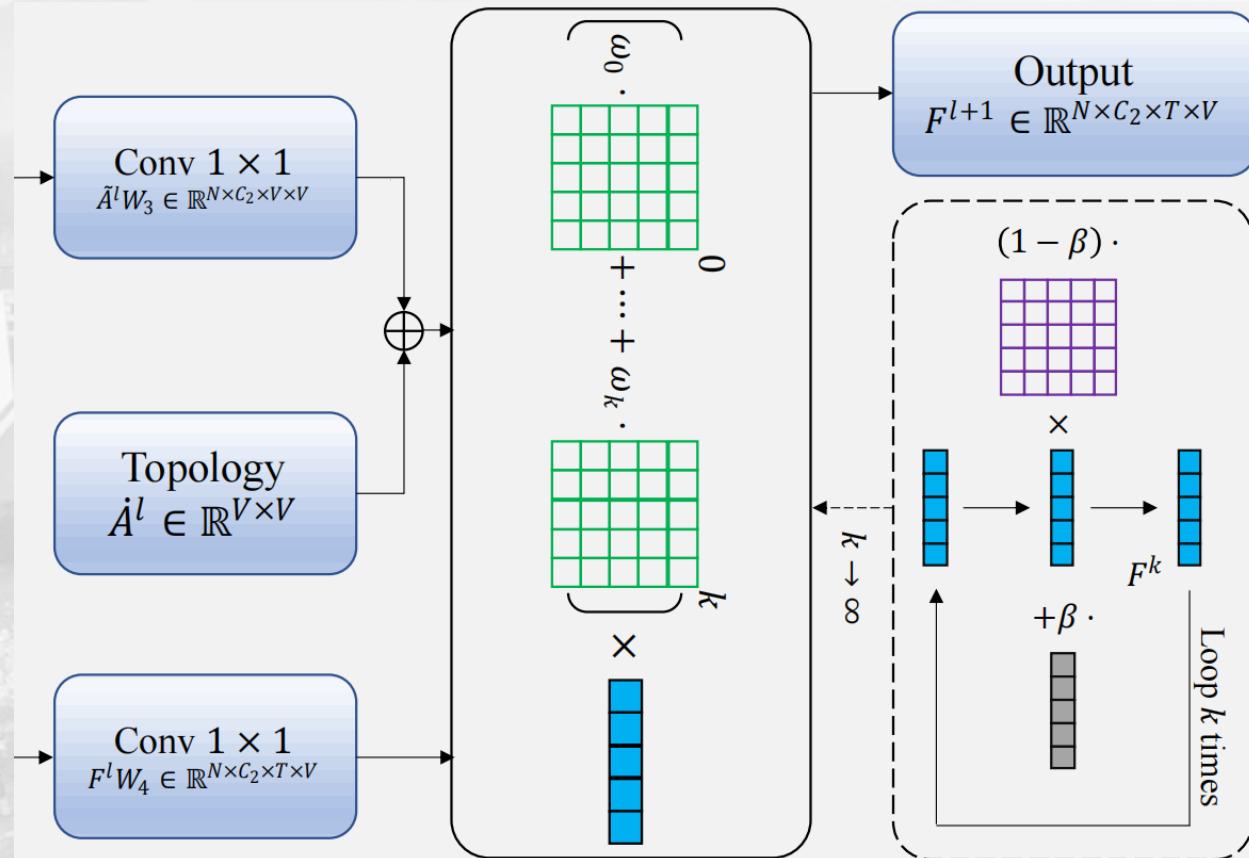
关系弱: $v1-v3$

关系不存在: $v1-v4$

```
# TODO: 使用einsum优化运算量
1个用法
def k_hop(self, A):
    # A: N, C, V, V
    N, C, V, _ = A.shape
    # A0: 1, 1, V, V
    A0 = torch.eye(V, dtype=A.dtype).to(A.device).unsqueeze(0).unsqueeze(0) * self.fuse_alpha

    A_power = torch.eye(V, dtype=A.dtype).to(A.device).unsqueeze(0).unsqueeze(0)
    for i in range(1, self.loop_times + 1):
        A_power = torch.einsum(*args: 'ncuv,ncvw->ncuw', A, A_power)
        A0 = A_power * (self.fuse_alpha * (1 - self.fuse_alpha) ** i) + A0

    return A0
```



$$F^{k+1} = (1 - \beta) \bar{A} F^k + \beta F^l$$

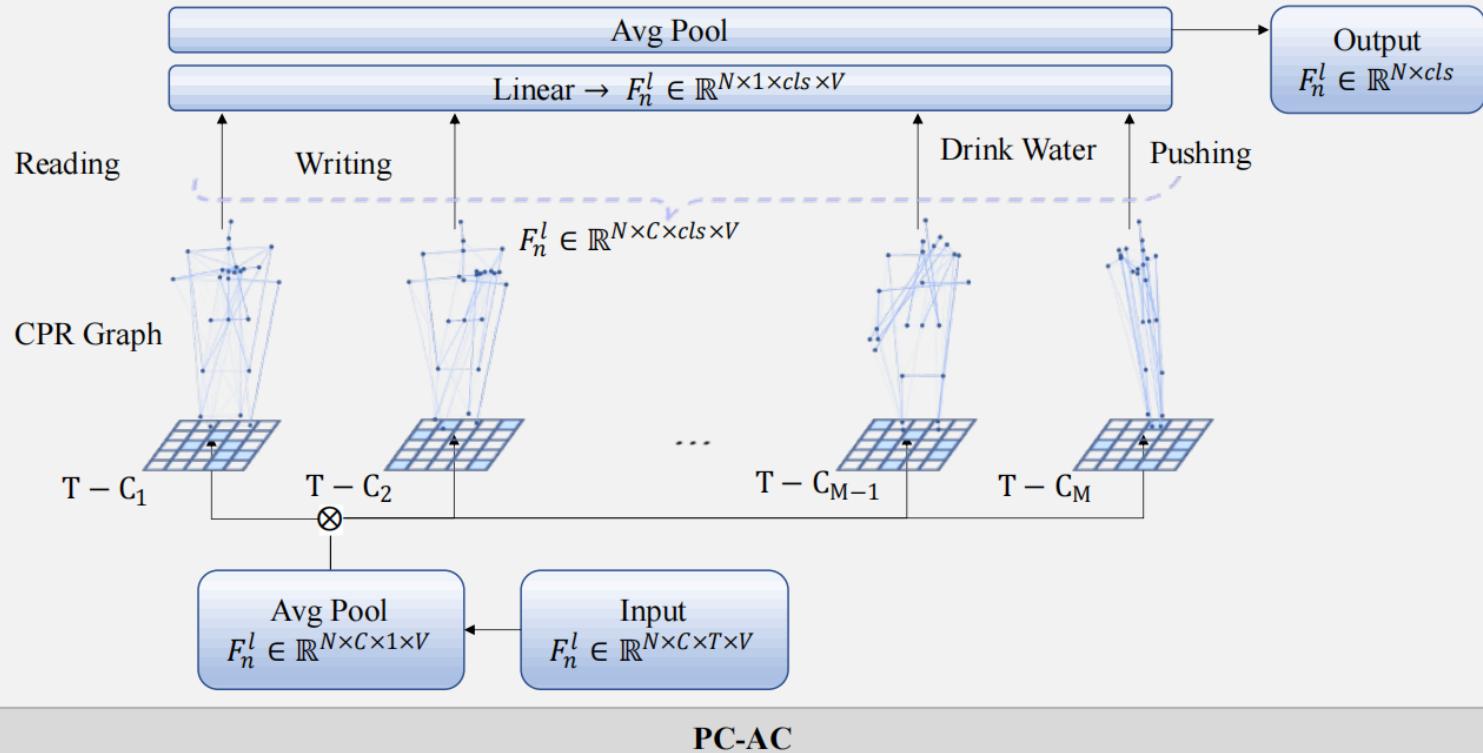
$$\lim_{K \rightarrow \infty} F^K = \bar{A} F^l$$

骨架序列包含固定数量的边 ε ，通过上述近似，我们得出了**多跳注意的复杂性与单跳注意没有太大区别**。复杂度都可以表示为 $O(|\varepsilon|)$

模型结构

XDU

PC-AC+A Priori Consistency-Assisted Classification Module



先验一致性辅助分类 (PC-AC)

利用**类关系拓扑图CPR**，包含与预测操作相关的先验节点关系，以帮助GCN执行**节点特征聚合**。

PC-AC模块在训练时向GCN添加了额外的**分支协同监督**，迫使每个特征通过一个包含特定类别拓扑的分支，从而为该类进行预测。

损失函数： the cross-entropy loss

$$L_{aux} = - \sum_k y_k \cdot \log(\hat{y}_k)$$

$$L_{pri} = - \sum_k y_k \cdot \log(\hat{y}_k)$$

$$\hat{y}_k = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

目标： $\arg \min_{\theta} (L(f_{\theta}^{pri}(x), \hat{y}) + \lambda L(f_{\theta_n}^{aux}(x), \hat{y}))$

4 Part Four

实验设计及结果

简要介绍实验的设计和实验结果，突出本篇论文的创新点

多流融合策略（关节、骨骼、关节运动、骨骼运动）

实验数据集：

NTU RGB+D. 数据集包含56,880个骨架动作序列，有60个类，可以分为日常行为、健康行为和交互行为。所有的动作数据都是由三个微软Kinect v2相机从不同的角度同时捕获的。本文提出了两种评价方案：

- 1) X-sub，其中训练集为20名受试者的数据，其余20名受试者的数据为测试集。
- 2) X-view，使用2号和3号摄像机捕获的数据作为训练集，使用1号摄像机捕获的数据作为测试集。

NTU RGB+D 120. 最常用的是增量数据集NTU RGB+D 120 的NTU RGB+D。它有120个类，106位受试者，和114,480个骨骼序列。所有这些行动都是由三个摄像机获得的。引入两个基准来评估NTU RGB+D 120：

- 1) X-sub，这与NTU RGB+D一样，需要区分两组受试者，每组由53名志愿者组成。
- 2) X-setup，其中获取不同配置的数据。训练集为偶数配置数据，测试集为奇数配置数据。

NW-UCLA. 数据集包含了10个基本的人类动作和来自10个演员的1494个视频剪辑。所有数据均来自同时捕获的三个Kinect相机。根据论文引入的评估方案，将前两个摄像机的数据作为训练集，将最后一个摄像机的数据作为测试集。

实验设计及结果

多流融合策略 (关节、骨骼、关节运动、骨骼运动)

XDU

TABLE 1: The comparison of Top-1 accuracy (%) on the NTU RGB+D [31] bench

—— UCLA [33] benchmark.

VA	25.5
ST	57.9
AS	61.8
2s-AGC	73.2
Direc	84.9
S	84.9
Shi	87.1
DC-G	87.6
Dyna	88.4
M	88.6
DI	88.8
MS	88.9
Effic	90.6
CT	90.7
Inf	90.7
LA	88.0

Methods	Top1
Lie Group [57]	74.2
Actionlet ensemble [58]	76.0
HBRNN-L [59]	78.5
Ensemble TS-LSTM [38]	89.2
AGC-LSTM [39]	93.3
4s Shift-GCN [40]	94.6
DC-GCN+ADG [54]	95.3
CTR-GCN [8]	96.5
InfoGCN [6]	97.0
LA-GCN (ours)	97.6

X-Set
25.5
57.9
61.8
73.2
84.9
84.9
87.1
87.6
88.4
88.6
88.8
88.9
90.6
90.7
90.7
88.0
90.9
91.3
91.8

TABLE 4: (i) The comparison of accuracy without and with PC-AC, where the λ of L_{aug} is 0.2. (ii) Contribution of different text prompts.

Methods	Top1
w/o L_{aug}	84.9
L_{total}	86.1 \uparrow 1.2
p1: [J] function in [C].	85.6 \uparrow 0.7
p2: What happens to [J] when a person is [C]?	85.8 \uparrow 0.9
p3: What will [J] act like when [C]?	86.1 \uparrow 1.2
p4: When [C][J] of human body.	85.5 \uparrow 0.6
p5: When [C] what will [J] act like?	85.7 \uparrow 0.8
p6: When a person is [C], [J] is in motion.	85.5 \uparrow 0.6

TABLE 7: λ selection of PC-AC with different text prompts T-C.

λ	0.1	0.2	0.3	0.5
p1: [J] function in [C].	85.0	85.6	85.3	84.8
p2: What happens to [J] when a person is [C]?	85.1	85.8	85.5	85.1
p3: What will [J] act like when [C]?	85.3	86.1	85.6	85.3
p4: When [C][J] of human body.	85.0	85.5	85.0	84.8
p5: When [C] what will [J] act like?	85.2	85.7	85.2	84.9
p6: When a person is [C], [J] is in motion.	85.1	85.5	85.2	85.0

TABLE 9: The accuracy (%) of the five-stream (5s) and six-stream (6s) ensemble on the NTU RGB+D 120 X-Sub split between the traditional four-stream (4s) and the new “bone” representatives. The basis is the joint acc: 86.5%. The model accuracy (%) of the new “bone” representations $\{B_{p1}, B_{p2}, B_{p3}, B_{p4}, B_{p5}, B_{p6}\}$ on the NTU RGB+D 120 X-Sub split are 85.9, 86.0, 84.9, 85.9, 85.8, and 85.8, respectively.

虽然从新的骨骼表示中学习到的模型表现不如原始的骨模型，但其在集成实验中的良好性能表明，新的“骨”的可变性可以有效地补充特征学习。

Modes	Components	Acc
2s	Joint + Bone	89.7 \uparrow 3.2
4s	J + B + JM + BM	89.9 \uparrow 3.4
5s	$4s + B_{p1}$	90.1 \uparrow 3.6
	$4s + B_{p2}$	90.4 \uparrow 3.9
	$4s + B_{p3}$	90.3 \uparrow 3.8
	$4s + B_{p4}$	90.1 \uparrow 3.6
	$4s + B_{p5}$	90.4 \uparrow 3.9
	$4s + B_{p6}$	90.2 \uparrow 3.7
6s	$4s + B_{p2} + B_{p5}$	90.7 \uparrow 4.2

- [1] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 7444–7452. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17135>
- [2] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 2019, pp. 12 026–12 035.
- [3] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel wise topology refinement graph convolution for skeleton-based action recognition," in 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, 2021, pp. 13 339–13 348.
- [4] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016. 2

XDU

THANK FOR ATTENTION

厚德 求真 励学 笃行