요약

- 제안한 것들
  - disentangled multi-scale aggregation scheme (MS)
    - 기존 논문들은 convolution에 사용할 hop이 커질수록 물리적으로 더 멀리 떨어진 joint와 convolution이 가능하나, 사용하는 normalize function+aggregation 특성상 먼 joint보다 가까운 joint에 대해서 가중치를 높게하는 biased weighting problem 존재
    - 그런 문제가 있음에도 먼 joint의 정보를 사용하는데 모델 성능이 향상되었으니 더 가중치를 높혀주자
  - unified spatial-temporal graph convolution (G3D)
    - 모션 데이터를 시간축/공간축으로 분해해서 모델이 연산하는데 일괄적으로 계산하도록 graph convolution에 사용할 adjacency matrix를 인접 프레임의 인접성도 포함시키자
  - 위 두개를 합친 MS-G3D 기반의 모델 제안
- 그림 1)
  - (a) : 기존논문에서는 공간축연산(빨간선) 후 시간축연산(파란선)하는 방식
  - (b) : G3D를 사용해서 시공간 연산을 동시에하기
  - (c) : G3D와 MS를 섞어서 거리가 달라도 동일한 가중치의 adjacency matrix제공

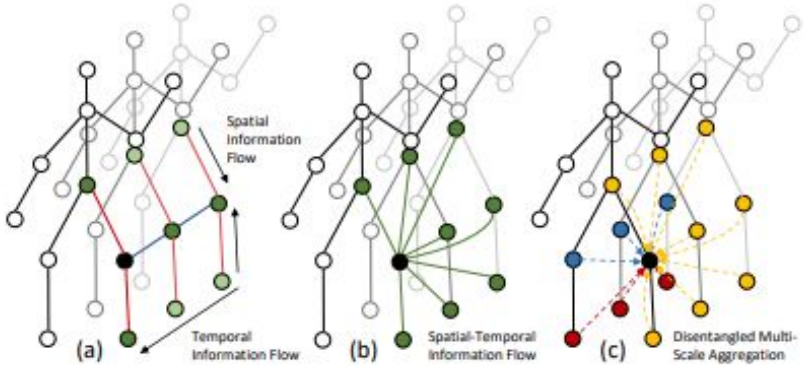


Figure 1: (a) Factorized spatial and temporal modeling on skeleton graph sequences causes *indirect* information flow. (b) In this work, we propose to capture cross-spacetime correlations with *unified* spatial-temporal graph convolutions. (c) Disentangling node features at separate spatial-temporal neighborhoods (yellow, blue, red at different distances, partially colored for clarity) is pivotal for effective multi-scale learning in the spatial-temporal domain.

Disentangled aggregation (multi-scale)

- 기존 논문들은 convolution에 사용할 hop이 커질수록 물리적으로 더 멀리 떨어진 joint와 convolution이 가능하나,
  사용하는 normalize function+aggregation 특성상 먼 joint보다 가까운 joint에 대해서 가중치를 높게하는 **biased weighting problem** 존재
  => 그림 (2) 윗줄 + 식 (2)
- 식 (2) : 기존 논문의 GCN. spatial partitioning(부모,자신,자식 joint별로 다른 label) 사용
  식 (2)의 adjacency matrix는 다양한 normalize 기법(Laplacian, random-walk, 등) 으로 weighted feature average
  - 이때 k-hop 보다 짧은 거리를 가진 joint는 먼 거리를 가진 joint 보다 여러 adjacency matrix에서 인접함이 표현되어 실질적으로 가까운 joint에 더 가중치가 높아지는 문제가 발생
- 본 논문은 동일한 가중치를 주기 위해 k거리만큼 떨어진 joint에 대해서만 인접함을 표현
  => 그림 (2) 아랫줄 + 식(3)
  - 같은 distance에 대해 같은 label을 적용 (distance partitioning과 유사함)
- 최종적으로 GCN에서 기존논문의 식(2)을 식(4)로 변경

$$\mathbf{X}_t^{(l+1)} = \sigma\left(\sum_{k=0}^{K} \hat{\mathbf{A}}^k \mathbf{X}_t^{(l)} \Theta_{(k)}^{(l)}\right), \qquad (2)$$

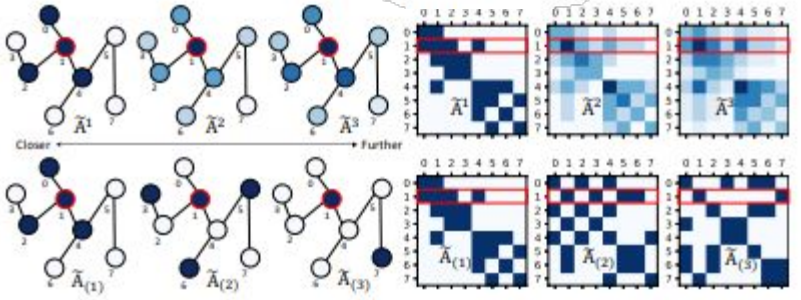


Figure 2: Illustration of the **biased weighting problem** and the proposed **disentangled aggregation scheme**. Darker color indicates higher weighting to the central node (red). **Top left**: closer nodes receive higher weighting from adjacency powering, which makes long-range modeling less effective, especially when multiple scales are aggregated. **Bottom left**: our proposed disentangled aggregation models joint relationships at each neighborhood while keeping identity features. **Right**: Visualizing the corresponding adjacency matrices. Node self-loops are omitted for visual clarity.

$$[\tilde{\mathbf{A}}_{(k)}]_{i,j} = \begin{cases} 1 & \text{if } d(v_i, v_j) = k, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \qquad (3)$$

$$\mathbf{X}_t^{(l+1)} = \sigma\left(\sum_{k=0}^{K} \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \tilde{\mathbf{A}}_{(k)} \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \mathbf{X}_t^{(l)} \Theta_{(k)}^{(l)}\right), \qquad (4)$$

Laplacian $\hat{\mathbf{A}} = \mathbf{L}^{\text{norm}} = \mathbf{I} - \mathbf{D}^{\frac{1}{2}}\mathbf{A}\mathbf{D}^{\frac{1}{2}}$;

random-walk normalized adjacency $\hat{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$

generally $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$

Unified Spatial-Temporal Modeling (G3D)

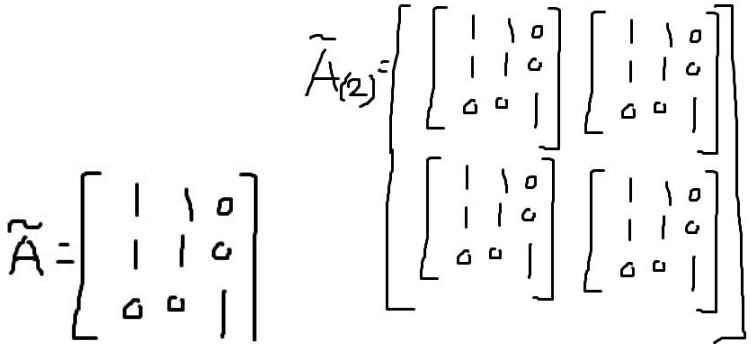
- 모션 데이터를 시간축/공간축으로 분해해서 모델이 연산하는데 일괄적으로 계산하도록 graph convolution에 사용할 adjacency matrix를 인접 프레임의 인접성도 포함시키자
- **Cross-Spacetime Skip Connections**
  - 본 논문은 공간축 GCN연산에 사용할 adjacency matrix에 시간축에 대한 정보를 끼워넣기 위해서 시간축에 대한 윈도우 사이즈 t(타우) 만큼 adjacency matrix를 붙여서 사용 => 식 (5)
    - 직관적으로 프레임에 따라 물리적인 연결이 변화하지 않으니 각 부분행렬(A~)은 동일한 값을 가짐
    - 부분행렬이 타우x타우로 배치되어 마치 시간축에 대해서 1-hop spatial neighbor로 여겨짐
      ex) 3개의 joint에서 1번 joint와 2번 joint가 연결되있는 인접행렬에 타우를 2로 설정할 경우 아래 그림과같음
  - 식(5)의 adjacency matrix를 활용하여 기존논문의 GCN을 식(6)으로 변경
- **Dilated Windows**
  - 멀리 떨어진 frame과의 인접성을 표현하기 위해서 dilation rate d를 추가
- Multi-Scale G3D
  - 이전의 disentangled aggregation과 G3D를 합치기 => 식 (7)
  - Mult-Scale의 인접행렬을 부분행렬로 하여 G3D의 인접행렬로써 연산

$$\tilde{\mathbf{A}}_{(\tau)} = \begin{bmatrix} \tilde{\mathbf{A}} & \cdots & \tilde{\mathbf{A}} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}} & \cdots & \tilde{\mathbf{A}} \end{bmatrix} \in \mathbb{R}^{\tau N \times \tau N}. \quad (5)$$

시간축 사이즈 / 사람수

$$[\mathbf{X}_{(\tau)}^{(l+1)}]_t = \sigma\left(\tilde{\mathbf{D}}_{(\tau)}^{-\frac{1}{2}}\tilde{\mathbf{A}}_{(\tau)}\tilde{\mathbf{D}}_{(\tau)}^{-\frac{1}{2}}[\mathbf{X}_{(\tau)}^{(l)}]_t\Theta^{(l)}\right). \quad (6)$$

$$[\mathbf{X}_{(\tau)}^{(l+1)}]_t = \sigma\left(\sum_{k=0}^{K}\tilde{\mathbf{D}}_{(\tau,k)}^{-\frac{1}{2}}\tilde{\mathbf{A}}_{(\tau,k)}\tilde{\mathbf{D}}_{(\tau,k)}^{-\frac{1}{2}}[\mathbf{X}_{(\tau)}^{(l)}]_t\Theta_{(k)}^{(l)}\right), \quad (7)$$



모델 구현

- 그림 3) 모델 구조.
  - (a) : STGC block - (b)를 활용한 전체 모델 구조
    - 논문은 r=3, 블록별 feature size={96,192,384}를 사용
    - 첫 STGC 블록을 제외한 STGC 블록에서는 MS-G3D 내의 Sliding Temporal Window 블록에서 stride=2를 사용
  - (b) : 본논문이 제안한 기법 기반의 STGC block
    - 점선표시는 모델 사이즈 여유에 따라 다른 타우 t,d를 사용하는 MS-G3D를 추가해서 동시에 다양한 시-공간 학습을 유도
    - G3D는 인접행렬 사이즈가 타우 t에 따라 비약적으로 커져서 hop size k를 키우기 힘듦 => 지역적인 학습만 이뤄짐
    - 큰 k를 쓰기 위해서 공간축/시간축을 순차적으로 연산하는 GCN+TCN 블록을 추가
  - (c) : 시간축에 대해서도 multi-scale적용(dilation을 달리하여 Conv) + 학습이 용이하게 Residual connection추가
  - (e) : 기존 논문들이 Adjacency matrix는 학습되도록하는게 성능향상됨을 증명했으니 인접행렬별로 learnable adjacency matrix를 추가

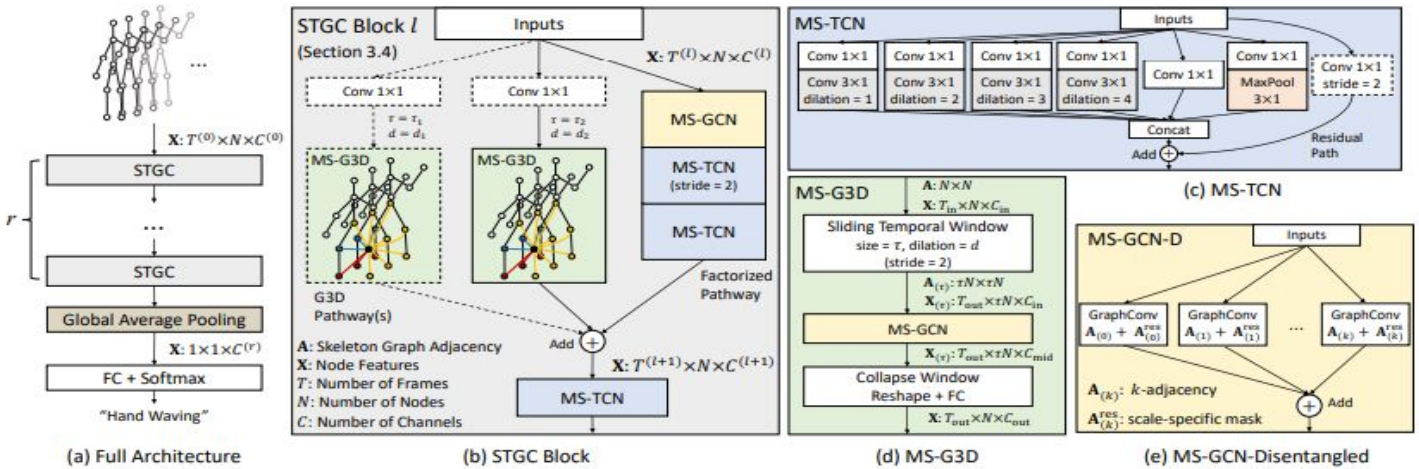


Figure 3: (Match components with colors) **Architecture Overview**. "TCN", "GCN", prefix "MS-", and suffix "-D" denotes temporal and graph convolutional blocks, and multi-scale and disentangled aggregation, respectively (Section 3.2). Each of the r STGC blocks (b) deploys a multi-pathway design to capture long-range and regional spatial-temporal dependencies simultaneously. **Dotted modules**, including extra G3D pathway, 1×1 conv, and strided temporal convolutions, are situational for model performance/complexity trade-off.

실험 결과

- NTU RGB+D 60 dataset으로 ablation study 진행
  - 표 1) K-hop에 따른 기법별 개별 정확도 실험 => STGC Block내에 GCN/G3D의 적절한 K찾기
    -E : 이전 논문들의 normalized adjacency matrix를 사용
    -D : 본 논문의 disentangled aggregation 사용
    Mask : learnable adjacency matrix 추가
    GCN / G3D : STGC block의 GCN/G3D 블록의 개별적인 학습 및 정확도 확인
    G3D의 타우 t= 5로 설정
    - Mask를 추가함으로써 predefined adjacency matrix에서 조절하는게 중요함을 보여줌
    - 일반적으로 GCN이 G3D보다 K값이 클때 성능이 좋아서 이후 학습에서는 GCN K=12, G3D K=5로 경험적으로 설정
  - 표 2) 2s-AGCN에서 joint stream과 성능 비교 및 적절한 G3D의 모델파라미터 찾기
    - 첫줄 : 기존 joint stream-AGCN 모델 / AGCN의 TCN => MS-TCN로 교체
    - 둘째줄 : 본 논문의 STGC block 중 MS-GCN 만 사용하기
      MS-GCN을 더깊게/넓게 하기 / 동일 모델 병렬로 배치하여 성능개선이 됨을 보임
    - 셋째줄 : MS-GCN + MS-G3D 구성에 모델파라미터 별 정확도 변화
    - 넷째줄 : MS-GCN + 2x MS-G3D 구성
      모델사이즈가 다른 모델과 비슷하도록 feature size 조절
  - 표 3) cross-spacetime - 식(5) -이 필요한지 다른 설정들과 성능 비교
    (1) : 원래 현재프레임의A를 유지한채, superdiagonal/subdiagonal 에는 단위행렬로 설정
    (2) : 현재프레임의A를 유지한채 나머지 모든 원소A를 단위행렬로 설정

Methods	Number of Scales			
	$K=1$	$K=4$	$K=8$	$K=12$
GCN-E	85.1	85.6	86.5	86.6
GCN-D	85.1	87.0	86.9	86.8
GCN-E + Mask	86.1	87.0	87.5	87.7
GCN-D + Mask	86.1	86.9	87.9	87.8
G3D-E	85.1	85.5	85.4	85.5
G3D-D	85.1	86.4	86.5	86.4
G3D-E + Mask	86.6	87.0	86.5	86.2
G3D-D + Mask	86.6	87.4	87.1	87.0

Table 1: Accuracy (%) with multi-scale aggregation on individual pathways of STGC blocks with different K. "Mask" refers to the residual masks $\mathbf{A}^{\text{res}}$. If $K>1$, GCN/G3D is **Multi-Scale** (**MS-**).

Model Configurations	Params	Acc (%)
Baseline (Js-AGCN [33])	3.5M	86.0
Baseline + MS-TCN	1.6M	86.7
MS-GCN (Factorized Pathway) Only	1.4M	87.8
with 2.5× Capacity	3.5M	88.5
with Dual Pathway	2.8M	88.6
MS-GCN (Factorized Pathway)		
with MS-G3D ($\tau=3, d=1$)	2.7M	89.0
with MS-G3D ($\tau=3, d=2$)	2.7M	89.1
with MS-G3D ($\tau=3, d=3$)	2.7M	89.1
with MS-G3D ($\tau=5, d=1$)	3.2M	89.2
with MS-G3D ($\tau=5, d=2$)	3.2M	89.2
with MS-G3D ($\tau=7, d=1$)†	3.0M	89.0
with 2 MS-G3D Pathways† $\tau=(3,3), d=(1,2)$	2.8M	89.3
with 2 MS-G3D Pathways† $\tau=(3,5), d=(1,1)$	3.2M	89.4

Table 2: Model accuracy with various settings. MS-GCN and MS-G3D uses $K \in \{12, 5\}$ respectively. †Output channels double at the collapse window layer (Fig. 3(d), C_{mid} to C_{out}) instead of at the graph convolution (C_{in} to C_{mid}) to maintain similar budget.

G3D Graph Connectivity	Params	Acc (%)
(1) Grid-like	2.7M	88.7
(2) Grid-like + dense self-edges	2.7M	88.6
(Eq. 5) Cross-spacetime edges	2.7M	89.1

Table 3: Comparing graph connectivity settings ($\tau=3, d=2$).

실험 결과

- 데이터셋 별 기존 모델들과 성능비교
  - 표 4) NTU RGB+D 120에서 비교
  - 표 5) NTU RGB+D 60에서 비교
  - 표 6) Kinetics에서 비교

Methods	NTU RGB+D 120	
	X-Sub (%)	X-Set (%)
ST-LSTM [26]	55.7	57.9
GCA-LSTM [27]	61.2	63.3
RotClips + MTCNN [16]	62.2	61.8
Body Pose Evolution Map [28]	64.6	66.9
2s-AGCN [33]	82.9	84.9
MS-G3D Net	**86.9**	**88.4**

Table 4: Classification accuracy comparison against state-of-the-art methods on the NTU RGB+D 120 Skeleton dataset.

Methods	NTU RGB+D 60	
	X-Sub (%)	X-View (%)
IndRNN [23]	81.8	88.0
HCN [20]	86.5	91.1
ST-GR [18]	86.9	92.3
AS-GCN [21]	86.8	94.2
2s-AGCN [33]	88.5	95.1
AGC-LSTM [34]	89.2	95.0
DGNN [32]	89.9	96.1
GR-GCN [8]	87.5	94.3
MS-G3D Net (Joint Only)	89.4	95.0
MS-G3D Net (Bone Only)	90.1	95.3
MS-G3D Net	**91.5**	**96.2**

Table 5: Classification accuracy comparison against state-of-the-art methods on the NTU RGB+D 60 Skeleton dataset.

Methods	Kinetics Skeleton 400	
	Top-1 (%)	Top-5 (%)
ST-GCN [50]	30.7	52.8
AS-GCN [21]	34.8	56.5
ST-GR [18]	33.6	56.1
2s-AGCN [33]	36.1	58.7
DGNN [32]	36.9	59.6
MS-G3D Net	**38.0**	**60.9**

Table 6: Classification accuracy comparison against state-of-the-art methods on the Kinetics Skeleton 400 dataset.