요약

- 2s-AGCN 저자의 후속 논문 ‘(2019)Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition’
- 제안기법
  - 이전 논문에서 제안한 adjacency matrix 수정
    - A,B,C matrix => B(초기값 A),C 로 바꾸고 C는 레이어 별 학습가능한 계수 알파a 와 곱해지는 게이트 매커니즘 제안
      - 학습 초기에 B를 학습시키는 것이 불안정하게 만들기 때문에
        (1) 초기 단계에서 B에 대한 학습을 차단하거나
        (2) 초기 단계에서 기존 논문의 A를 놔두고 B,C에 대한 계수를 0으로 초기화해서 차단하기
      - 실험을 통해서 방법 (1)이 더 성능이 잘 나오는 것을 확인
  - AGCN 블록 가운데에 STC-attention module을 넣은 Multi-Stream Attention-enhanced Adaptive Graph Convolutional neural Network(MS-AAGCN) 제안
    - ‘(2020)PGCN-TCA: Pseudo Graph Convolutional Network With Temporal and Channel-Wise Attention for Skeleton-Based Action Recognition’ 논문의 시/공간 Attention과 유사함
    - 본 논문은 시간/공간/채널 별로 Attention을 계산하고 Residual connection으로 원본과 더하는 STC-Attention 모듈 제안
    - 실험을 통해 병렬로 계산해서 적용하는것보다 순차적인 계산/적용이 성능이 더 잘나오는 것을 확인
  - 더 많은 스트림 사용
    - 기존 논문 : Joint-stream, Bone-Stream
    - 본 논문 : 기존 논문 + 동일 Joint의 이전프레임과의 차이(Joint-M) + 동일 Bone의 이전프레임과의 차이(Bone-M)
- 실험에서 RGB 입력 기반의 모델과 앙상블해서 높은 정확도를 보여줌
  - “Gesture recognition using spatiotemporal deformable convolutional representation” 논문의 모델 + 배경 날려주기 + 본논문모델 앙상블

Adaptive Graph Convolution

- 이전 논문에서 제안한 adjacency matrix 수정 - **그림 (2)**
  - A,B,C matrix => B(초기값 A),C 로 바꾸고 C는 레이어 별 학습가능한 계수 알파a 와 곱해지는 게이트 매커니즘 제안
    - 학습 초기에 B를 학습시키는 것이 불안정하게 만들기 때문에
      (1) 초기 단계에서 B에 대한 학습을 차단하거나
      (2) 초기 단계에서 기존 논문의 A를 놔두고 B,C에 대한 계수를 0으로 초기화해서 차단하기
    - 실험을 통해서 방법 (1)이 더 성능이 잘 나오는 것을 확인
  - 식 2,3) ST-GCN의 GCN 계산 식 - Learnable Mask + Physically Adjacency Matrix
  - 식 4) ST-GCN의 식(2)를 변경한 논문 기법 - Learnable Global B + Attention Score C
  - 식 5) C 계산 방식 - Query/key 계산 + dot product + Softmax로 attention score 계산
  - 식 6) 식 (5)에 대해서 풀어서 표현

$$\hat{\mathcal{X}}_k = |\mathbf{M}_k \odot \mathbf{A}_k \mathbf{X}_{(3)}|_{C_{in} \times T \times N} \quad (2)$$

$$\mathcal{Y} = \sum_k^{K_v} |\mathbf{W}_k \hat{\mathbf{X}}_{k(1)}|_{C_{out} \times T \times N} \quad (3)$$

$$\hat{\mathcal{X}}_k = |(\mathbf{B}_k + \alpha \mathbf{C}_k)\mathbf{X}_{(3)}|_{C_{in} \times T \times N} \quad (4)$$

$$f(v_i, v_j) = \frac{e^{\theta(v_i)^\dagger \phi(v_j)}}{\sum_{j=1}^{N} e^{\theta(v_i)^\dagger \phi(v_j)}} \quad (5)$$

$$\mathcal{M}_{\theta k} = |\mathbf{W}_{\theta k} \mathbf{X}_{(1)}|_{C_e \times T \times N}$$

$$\mathcal{M}_{\phi k} = |\mathbf{W}_{\phi k} \mathbf{X}_{(1)}|_{C_e \times T \times N}$$

$$\mathbf{C}_k = SoftMax(\mathbf{M}_{\theta k(3)} \mathbf{M}_{\phi k}{}^{\dagger}_{(3)}) \quad (6)$$

$\mathcal{X} \in \mathbb{R}^{C_{in} \times T \times N}$ $\mathbf{X}_{(3)}$, is an $N \times C_{in}T$

$|\mathbf{X}_{(3)}|_{C_{in} \times T \times N}$ is the original $C_{in} \times T \times N$

$\mathbf{M}_k \in \mathbb{R}^{N \times N}$

$\mathbf{W}_\theta, \mathbf{W}_\phi \in \mathbb{R}^{C_e \times C_{in}}$

$\mathbf{C}_k \in \mathbb{R}^{N \times N}$

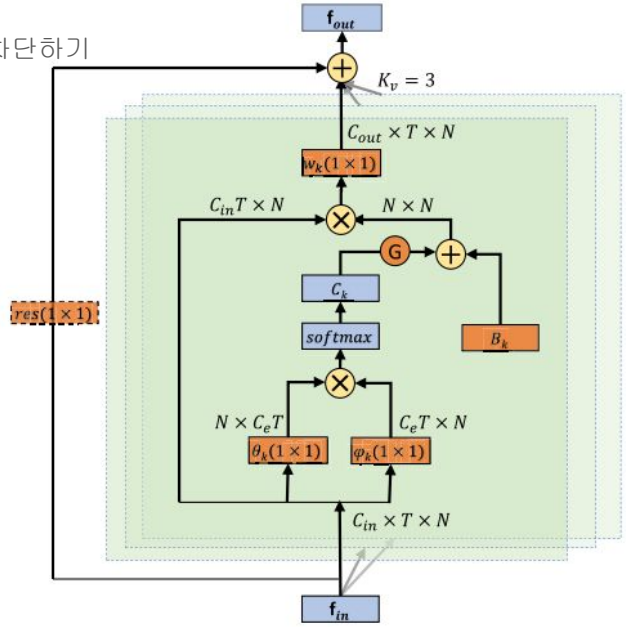

Fig. 2. Illustration of the adaptive graph convolutional layer (AGCL). There are two kinds of graphs in each layer, i.e., $\mathbf{B}_k$ and $\mathbf{C}_k$. The orange box indicates that the part is the parameter of the network and is updated during the training process. θ and ϕ are two embedding functions whose kernel size is (1×1). K_v denotes the number of subsets. $\oplus$ denotes the element-wise addition. $\otimes$ denotes the matrix multiplication. G is the gate that controls the importance of the two kinds of graphs. The residual box (dotted line) is only needed when C_{in} is not the same as C_{out}.

STC-Attention Module

- AGCN 블록 가운데에 STC-attention module을 넣은 Multi-Stream Attention-enhanced Adaptive Graph Convolutional neural Network(MS-AAGCN) 제안 **=> 그림 3**
  - '(2020)PGCN-TCA: Pseudo Graph Convolutional Network With Temporal and Channel-Wise Attention for Skeleton-Based Action Recognition' 논문의 시/공간 Attention과 유사함
  - 본 논문은 시간/공간/채널 별로 Attention을 계산하고 Residual connection으로 원본과 더하는 STC-Attention 모듈 제안
  - 실험을 통해 병렬로 계산해서 적용하는것보다 순차적인 계산/적용이 성능이 더 잘나오는 것을 확인
- 그림 3) - STC attention 모듈 그림
  - 학습을 안정화하기위해 residual connection 사용
  - 공간/시간/채널 별 Attention 계산 모듈이 있음
  - 왜 썼는지에 대한 설명은 없음
- 식 7-9) 공간/시간/채널 어텐션별 수식

$$\mathcal{X} \in \mathbb{R}^{C \times T \times N}$$

$$\mathcal{M}_s \in \mathbb{R}^{1 \times 1 \times N} \quad \mathcal{M}_t \in \mathbb{R}^{1 \times T \times 1} \quad \mathcal{M}_c \in \mathbb{R}^{C \times 1 \times 1}$$

$$\mathcal{M}_s = \sigma(g_s(AvgPool_t(\mathcal{X})) \tag{7}$$

$$\mathcal{M}_t = \sigma(g_t(AvgPool_s(\mathcal{X})) \tag{8}$$

$$\mathcal{M}_c = \sigma(g_{c2}(\delta(g_{c1}(AvgPool_{st}(\mathcal{X}))))) \tag{9}$$

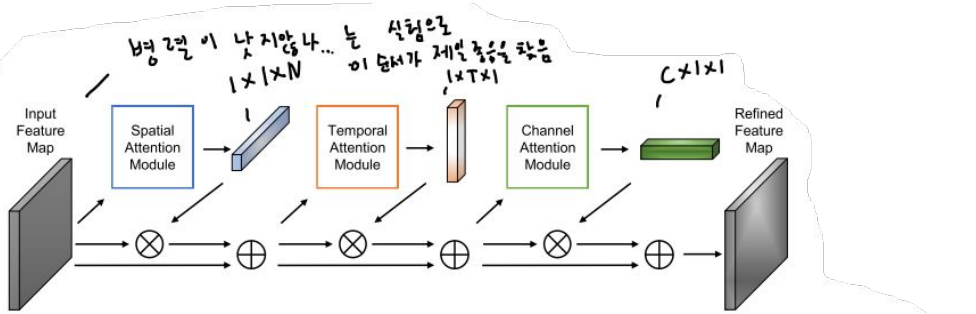

Fig. 3. Illustration of the STC-attention module. Three sub-modules are arranged in the orders of SAM, TAM and CAM. ⊗ denotes the element-wise multiplication. ⊕ denotes the element-wise addition. The generated attention maps are multiplied to the original feature maps. A residual connection is added for all attention modules to stabilize the training.

Network Architecture / Multi-stream Framework

- Network Architecture
  - 그림 4) MS-AAGCN의 기본 블록
    - ConvS - 논문이 제안한 GCN
    - ConvT - ST-GCN과 같이 시간축에 대한 convolution (시간축 커널길이 x 1)
  - 그림 5) 본논문의 네트워크 구조
    - ST-GCN과 같은 블록개수
- Multi-stream Framework => 그림 6
  - 이전 논문 : Joint-stream, Bone-Stream
  - 본 논문
    - 이전 논문의 2-stream
    - 동일 Joint의 이전프레임과의 차이(Joint-M)
    - 동일 Bone의 이전프레임과의 차이(Bone-M)

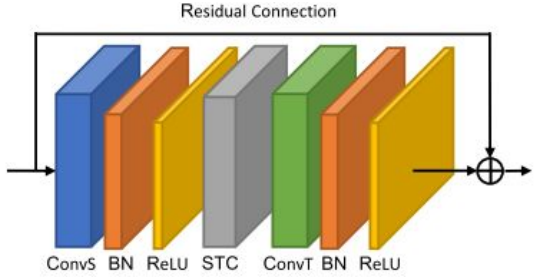

Fig. 4. Illustration of the basic block. ConvS represents the spatial AGCL, and ConvT represents the temporal AGCL, both of which are followed by a BN layer and a ReLU layer. STC represents the STC-attention module. Moreover, a residual connection is added for each block.

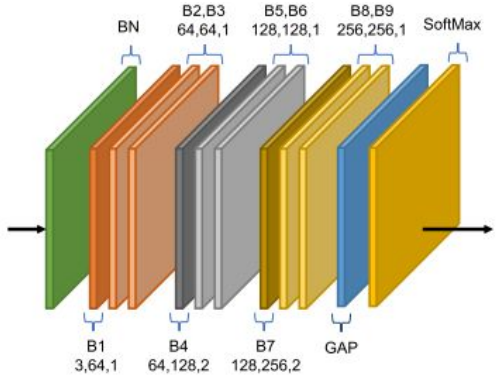

Fig. 5. Illustration of the network architecture. There are a total of 9 basic blocks (B1-B9). The three numbers of each block represent the number of input channels, the number of output channels and the stride, respectively. GAP represents the global average pooling layer.

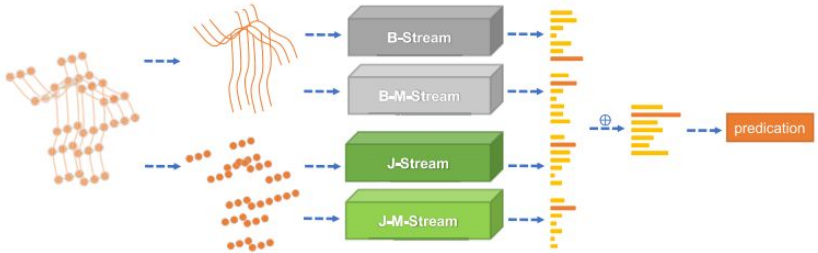

Fig. 6. Illustration of the overall architecture of the MS-AAGCN. The $SoftMax$ scores of the four streams are fused using weighted summation to obtain the final prediction. J denotes the joint information. B denotes the bone information. M denotes the motion information.

실험 결과 - ablation study

- a) learning rate scheduler + Data preprocessing => 그림(8) + 표 (1)
  - 기존 ST-GCN 성능 개선하기
  - ST-GCN의 learning rate scheduler 변경 => 표 (1) Before preprocessing
  - 그림(8)-왼쪽 처럼 다른 시점의 카메라 뷰에 대한 데이터를 똑바로 세워주는 preprocessing 진행
    - right shoulder(5번째 관절)에서 left shoulder(9번째 관절)까지 3D 벡터에 평행한 X축 고정 + spine base(21번)에서 spine(2번)까지 Y축 고정
  - 데이터에 노이즈가 많더라~
- b) adjacency matrix 비교 => 표 (2)
  - 표 (1)의 개선된 ST-GCN을 baseline으로 선정
  - 어떤 adjacency matrix가 학습에 좋을까 비교실험
  - AGCN-A : ST-GCN에서 learnable Mask 제거 => Mask가 성능에 영향이 있음
  - AGCN-B / AGCN-C : 논문이 제안한 B/C만 썼을 때 결과
  - AGCN-ABC : 초기에 A만 사용하도록 B,C 계수를 0으로 초기화
  - AGCN-BC : 본 논문이 제안한 B+C
  - AGCN-BC-G : C에 게이트 메커니즘 추가

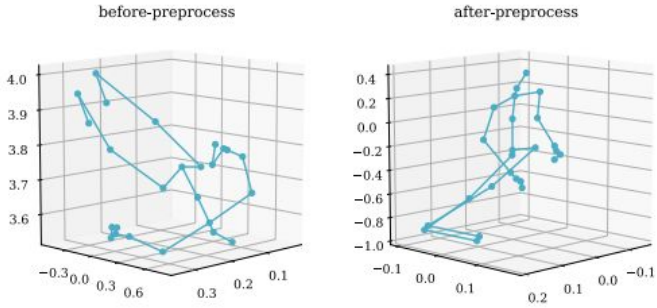

Fig. 8. Example of the data preprocessing on the NTU-RGBD dataset. The left is the original skeleton, and the right is the preprocessed skeleton.

TABLE I

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE USING REARRANGED LEARNING-RATE SCHEDULER AND DATA PREPROCESSING

Methods	CS (%)	CV (%)
original performance in [10]	81.5	88.3
before preprocessing	82.4	90.1
after preprocessing	84.3	92.7

TABLE II

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE ON THE NTU-RGBD DATASET. STGCN* DENOTE THE RESULT OF "AFTER PREPROCESSING" IN TAB. I. *A* DENOTES THE ADJACENCY MATRIX OF THE BODY-BASED GRAPH SHOWN IN EQ. 2. *B* AND *C* DENOTE THE GLOBAL GRAPH AND THE INDIVIDUAL GRAPH INTRODUCED IN SEC IV-A, RESPECTIVELY. *G* DENOTES USING THE GATING MECHANISM

Methods	CS (%)	CV (%)
STGCN*	84.3	92.7
AGCN-A	83.7	91.1
AGCN-B	86.4	93.6
AGCN-C	86.1	93.5
AGCN-ABC	86.6	93.7
AGCN-BC	87.0	94.1
AGCN-BC-G	87.4	94.4

실험 결과 - ablation study (계속)

- Attention module => **표 (3)**
  - 각 하위 어텐션 모듈이 성능개선이 되는건지 확인
  - ASTGCN-S / ASTGCN-T / ASTGCN-C : ST-GCN에 전체 STC를 개별적으로 적용한 결과
  - '-ADD' : 병렬로 처리해서 적용하기
  - '-STC' : 순차적으로 적용하기
  - AGCN : 표 (2)에서 최고성능을 보인 AGCN-BC-G
- Multi-Stream Framework => **표 (4)**
  - J-AAGCN : 표 (3)에서 최고성능을 보인 AAGCN-STC
  - 'B-' : bone-stream
  - 'J-M' : Joint Motion-stream
  - 'B-M' : Bone Motion-stream
  - 'JB' : 이전 논문과 같은 joint/bone stream 사용
  - 'MS' : 4개의 스트림 다 사용

TABLE III

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE ON THE NTU-RGBD DATASET FOR EACH OF THE ATTENTION SUB-MODULES AND DIFFERENT ARRANGEMENT STRATEGIES. STGCN* AND AGCN DENOTE THE STGCN* AND AGCN-BC-G IN TAB. II, RESPECTIVELY. STC MEANS CONCATENATING THE THREE SUB-MODULES SEQUENTIALLY

Methods	CS (%)	CV (%)
STGCN*	84.3	92.7
ASTGCN-S	86.1	93.1
ASTGCN-T	86.6	93.6
ASTGCN-C	86.5	93.7
ASTGCN-ADD	86.8	94.1
ASTGCN-STC	87.1	94.2
AGCN	87.4	94.4
AAGCN-ADD	87.7	94.8
AAGCN-STC	88.0	95.1

TABLE IV

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE WITH DIFFERENT INPUT STREAMS ON THE NTU-RGBD DATASET. JB REPRESENTS USING THE JOINT STREAM AND BONE STREAM. MS REPRESENTS USING ALL OF THE FOUR STREAMS. AAGCN DENOTES THE AAGCN-STC IN TAB. III

Methods	CS (%)	CV (%)
J-AAGCN	88.0	95.1
B-AAGCN	88.4	94.7
J-M-AAGCN	85.9	93.0
B-M-AAGCN	86.0	93.1
JB-AAGCN	89.4	96.0
MS-AAGCN	90.0	96.2

TABLE VI

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE WITH STATE-OF-THE-ART METHODS ON THE KINETICS-SKELETON DATASET. J, B AND M DENOTE USING THE JOINT INFORMATION, BONE INFORMATION AND MOTION INFORMATION, RESPECTIVELY. JB REPRESENTS USING THE JOINT AND BONE STREAM. MS REPRESENTS USING ALL OF THE FOUR STREAM

Methods	Top-1 (%)	Top-5 (%)
Feature Enc. [17]	14.9	25.8
Deep LSTM [14]	16.4	35.3
TCN [20]	20.3	40.0
ST-GCN [10]	30.7	52.8
ASGCN [27]	34.8	56.5
AGCN [16]	36.1	58.7
J-AAGCN (ours)	36.0	58.4
B-AAGCN (ours)	34.7	57.5
J-M-AAGCN (ours)	31.7	54.6
B-M-AAGCN (ours)	29.7	50.0
JB-AAGCN (ours)	37.4	60.4
MS-AAGCN (ours)	37.8± 0.084	61.0± 0.132

TABLE VII

THE ANALYSIS FOR THE PARAMETERS AND THE RUNTIME PERFORMANCE OF THE PROPOSED COMPONENTS. T DENOTES THE TIME REQUIRED FOR 100 MODEL RUNS USING CPU/GPU

Methods	#Params(M)	GFLOPS	T(ms) on CPU/GPU
ST-GCN	3.12	8.37	107/1.0
AGCN	3.47	9.34	145/2.7
AAGCN	3.78	9.35	156/3.2

실험 결과 - 성능비교

- 기존 논문들과 성능비교 => **표 (5-6)**
  - 표 5) NTU-RGBD dataset에 대한 성능비교
    - preprocessing도 포함된 성능향상
  - 표 6) Kinetics dataset에 대한 성능비교
- 런타임 비교 => **표(7)**
  - Intel(R) E5-2630 CPU (2.20GHz) and TITAN XP GPU
  - 1 batch size x 3 channel x 300 frame x 25 joint에 대한 연산 결과

TABLE V

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE WITH STATE-OF-THE-ART METHODS ON THE NTU-RGBD DATASET. CS AND CV DENOTE THE CROSS-SUBJECT AND CROSS-VIEW BENCHMARKS, RESPECTIVELY. WE COMPARED OUR METHODS WITH FOUR TYPES OF METHODS THAT ARE SEPARATED WITH A HORIZONTAL LINES: HANDCRAFT-FEATURE-BASED METHODS, RNN-BASED METHODS, CNN-BASED METHODS AND GCN-BASED METHODS (FROM TOP TO BOTTOM)

Methods	CS (%)	CV (%)
Lie Group [6]	50.1	82.8
HBRNN [7]	59.1	64.0
Deep LSTM [14]	60.7	67.3
ST-LSTM [49]	69.2	77.7
STA-LSTM [8]	73.4	81.2
VA-LSTM [18]	79.2	87.7
Ind-RNN [9]	81.8	88.0
SRN+TSL [19]	84.8	92.4
TCN [20]	74.3	83.1
Clips+CNN+MTLN [50]	79.6	84.8
Synthesized CNN [21]	80.0	87.2
CNN+Motion+Trans [22]	83.2	89.3
3scale ResNet152 [23]	85.0	92.3
ST-GCN [10]	81.5	88.3
DPRL+GCNN [25]	83.5	89.8
ASGCN [27]	86.8	94.2
AGCN [16]	88.5	95.1
AGC-LSTM [26]	89.2	95.0
MS-AAGCN (ours)	90.0± 0.109	96.2± 0.095

TABLE VI

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE WITH STATE-OF-THE-ART METHODS ON THE KINETICS-SKELETON DATASET. J, B AND M DENOTE USING THE JOINT INFORMATION, BONE INFORMATION AND MOTION INFORMATION, RESPECTIVELY. JB REPRESENTS USING THE JOINT AND BONE STREAM. MS REPRESENTS USING ALL OF THE FOUR STREAM

Methods	Top-1 (%)	Top-5 (%)
Feature Enc. [17]	14.9	25.8
Deep LSTM [14]	16.4	35.3
TCN [20]	20.3	40.0
ST-GCN [10]	30.7	52.8
ASGCN [27]	34.8	56.5
AGCN [16]	36.1	58.7
J-AAGCN (ours)	36.0	58.4
B-AAGCN (ours)	34.7	57.5
J-M-AAGCN (ours)	31.7	54.6
B-M-AAGCN (ours)	29.7	50.0
JB-AAGCN (ours)	37.4	60.4
MS-AAGCN (ours)	37.8± 0.084	61.0± 0.132

TABLE VII

THE ANALYSIS FOR THE PARAMETERS AND THE RUNTIME PERFORMANCE OF THE PROPOSED COMPONENTS. T DENOTES THE TIME REQUIRED FOR 100 MODEL RUNS USING CPU/GPU

Methods	#Params(M)	GFLOPS	T(ms) on CPU/GPU
ST-GCN	3.12	8.37	107/1.0
AGCN	3.47	9.34	145/2.7
AAGCN	3.78	9.35	156/3.2

실험 결과 - 시각화

- 그림 9-12) Adaptive graph에 대한 시각화
- 그림 9) - 레이어 별 global adjacency matrix인 B의 커널별(=라벨별) 값 시각화
  - 첫열 : NTU-RGBD 에서 물리적 연결을 표현
  - 윗줄 : 자식노드 라벨에 대한 인접행렬
  - 아랫줄 : 부모노드 라벨에 대한 인전행렬
  - 물리적 연결이 최적의 형태가 아님을 시각화
  - 상위 레이어에 포함된 정보가 더 semantic해서 하위레이어보다 많이 변경됨
- 그림 10) - 입력 액션에 따른 레이어 별 C 시각화
  - 윗줄 : 셀카 찍기
  - 아랫줄 : 던지기
- 그림 11) - 레이어별로 게이트메커니즘값 시각화
  - 상위 레이어에서는 C가 더 중요해짐을 보여줌

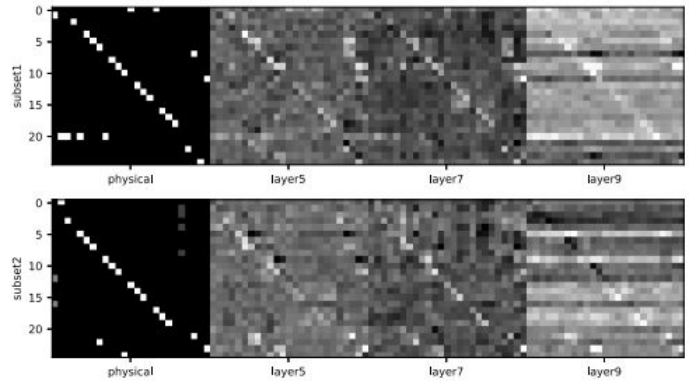

Fig. 9. Example of the learned adjacency matrices of the global graph. Different rows shows different subsets. The first column is the adjacency matrices of the human-body-based graph in the NTU-RGBD dataset. Others are examples of the learned adaptive adjacency matrices of different layers learned by our model.

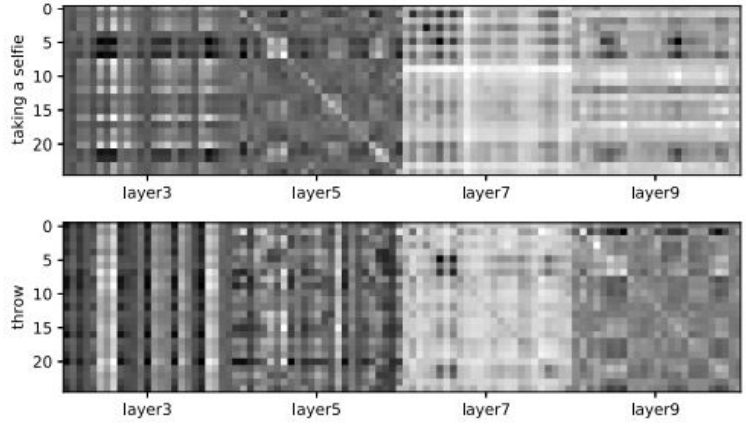

Fig. 10. Examples of the learned adjacency matrices of the individual graph. The first and second rows show different samples. Different columns represent different layers.

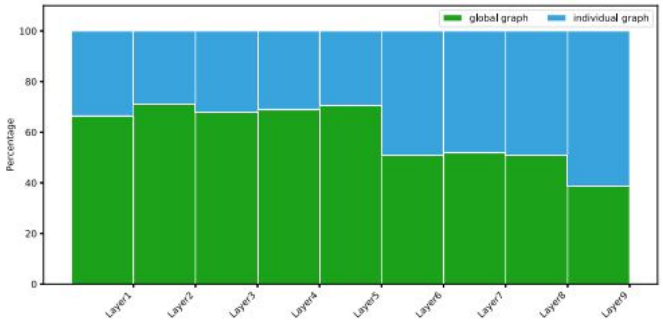

Fig. 11. Visualization for the importance of the two kinds of graphs in each of the layers.

실험 결과 - 시각화

- 그림 12) 입력 액션에 따른 레이어별 B/C 합친 결과
  - 주황색은 값이 높은 상위 50개
- 그림 13-14) Attention module 관련 시각화
- 그림 13) 입력 액션에 따른 레이어별 spatial attention 시각화 (joint크기에 따라 중요도표현)
  - 하위 레이어는 receptive field가 작아서 어텐션이 잘안됨
  - 상위 레이어로 갈수록 머리/손에 집중되는 것을 보여줌
- 그림 14) 입력 액션에 따른 각 레이어의 temporal attention 시각화
  - 셀카찍기-파랑/주황 : 손을 드는동안 + 마지막 자세취하는 프레임에 집중
  - 던지기 - 초록 : 셀카찍기랑 반대로 손이 내려가있는데 집중됨 (??)

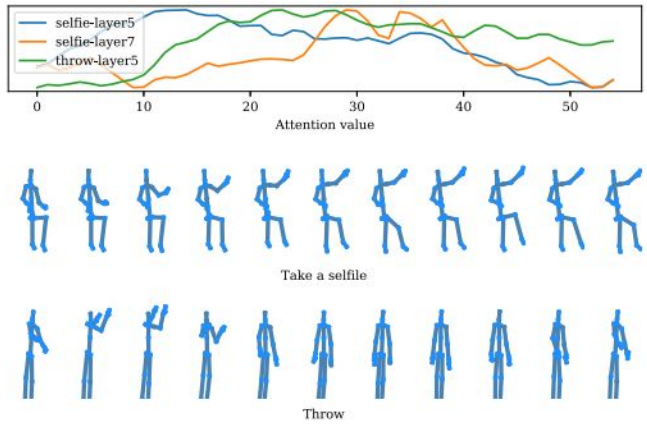

Fig. 14. Visualization of the temporal attention map. The first row shows the learned temporal attention weights for each of the frames for different layers and samples. The second and third rows show the corresponding skeleton sketches.

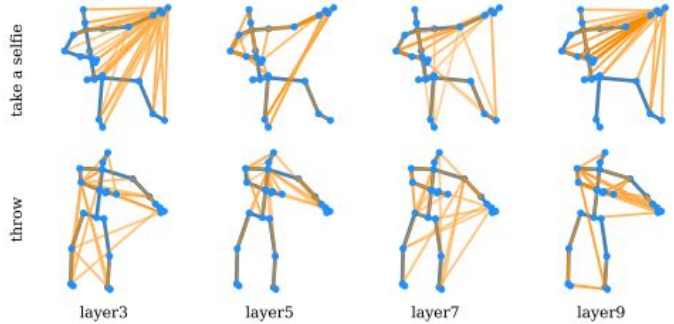

Fig. 12. Examples of the learned graph topologies. The orange lines represent the connections whose values are in the top 50.

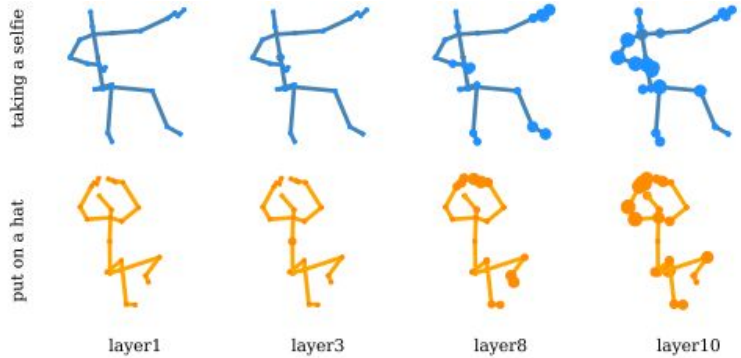

Fig. 13. Examples of the learned spatial attention maps. The size of the circle represents the importance of the corresponding joint.

실험 결과 - 다른기법과 곁들이기 + 시각화

- 그림 15-16) 도대체 뭐가 안되는건지 시각화
- 그림 15) Skeleton 기반 모델의 정확도 - RGB 기반 모델의 정확도를 시각화
  - RGB보다 못하는것들이 있다
  - 그림 16) 읽기/쓰기의 경우 스켈레톤은 손등 까지만 표현되있지만 RGB는 자세를 보고 파악할수있음
- 표 8) RGB 기반 모델과 본논문의 Skeleton 기반 모델을 앙상블했을 때 결과
  - '-C' : 이미지에서 인물빼고 배경날려버리는방법 사용
  - RNX3D101 : "Gesture recognition using spatiotemporal deformable convolutional representation" 논문의 ResNeXt3D-101 model 사용

Fig. 16. Two examples for class "reading" and class "writing". The white lines represent the skeletons. The two examples are hard to distinguish with only skeletons.

TABLE VIII

COMPARISONS OF THE ACTION RECOGNITION PERFORMANCE USING RGBS ON THE NTU-RGBD DATASET. CS AND CV ARE TWO BENCHMARKS. C DENOTE USING THE POSE-GUIDED CROPPING STRATEGY

Methods	Pose	RGB	CS (%)	CV (%)
DSSCA-SSLM [52]	✓	✓	74.9	-
Chained Network [53]	✓	✓	80.8	-
RGB + 2D Pose [54]	✓	✓	85.5	-
Glimpse Clouds [55]		✓	86.6	93.2
PEM [56]	✓	✓	91.2	95.3
I3D+RNN+Attention [57]	✓	✓	93.0	95.4
MS-AAGCN	✓		90.0	96.2
RNX3D101		✓	85.3	92.6
RNX3D101-C		✓	94.6	97.0
RNX3D101+MS-AAGCN	✓	✓	95.5	98.0
RNX3D101+MS-AAGCN-C	✓	✓	96.1	99.0

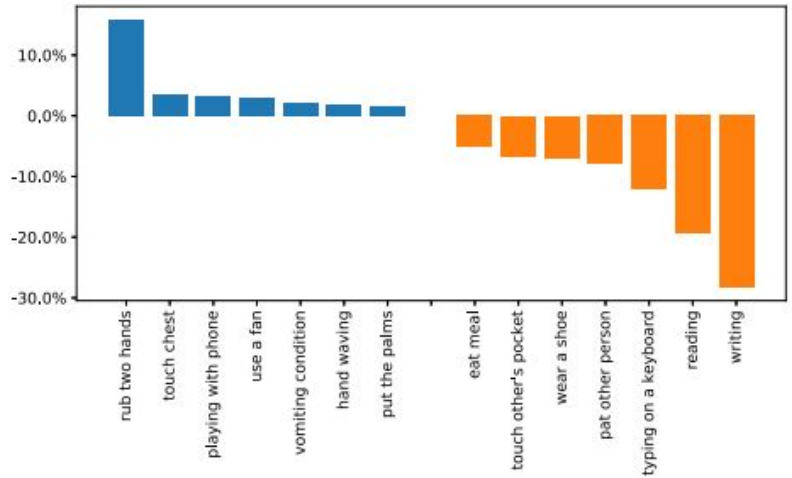

Fig. 15. Accuracy difference between the skeletons and the RGBs for different classes, i.e., ACC(skeleton)-ACC(RGB).