요약

- GCN을 활용한 많은 논문들이 있는데 너무 FLOPS가 높다 (데이터 한개 연산에 15 GFLOPS가 넘는다)
  - ST-GCN의 경우 16.2 GFLOPS/Action 발생 (4.0 GFLOPS 가량의 공간축 GCN + 12.2 GFLOPS 가량의 시간축 GCN)
- Convolution 할때 FLOPS를 줄이기위해 Shift Convolution을 하듯 GCN에도 Shifting을 해서 Convolution 하자 (Shift-GCN)
  - 발생 문제 1) 모델 설계단계에서 휴리스틱하게 정한 커널사이즈는 지역적인 학습만 되어 관절 간 다양한 관계를 학습하는데 한계가 있음
    발생 문제 2) shift를 진행하면서 다른 노드에 대한 feature 정보를 손실시킬 수도 있음
  - 위 문제를 해결하기 위해 local shift graph와 함께 모든 노드가 연결돼있는 것을 가정한 non-local shift graph operation도 혼용할 것을 제안
- 기존 논문들은 시간축에 대해서 Convolution할 때 커널 사이즈를 고정시켜서 모델 설계를 함
  - 발생 문제 1) 레이어마다 다양한 receptive field가 필요할 수 도있음
    발생 문제 2) 데이터 셋 별로 다른 시간축 receptive field가 필요할 수도 있음
  - 공간축 뿐 아니라 시간축에 대한 Conv에서도 learnable shift parameter를 추가해서 데이터셋에 따라 적응하는 shift기법을 설계
- FLOPS를 낮추면서 성능도 향상됨을 보임 => 그림 1

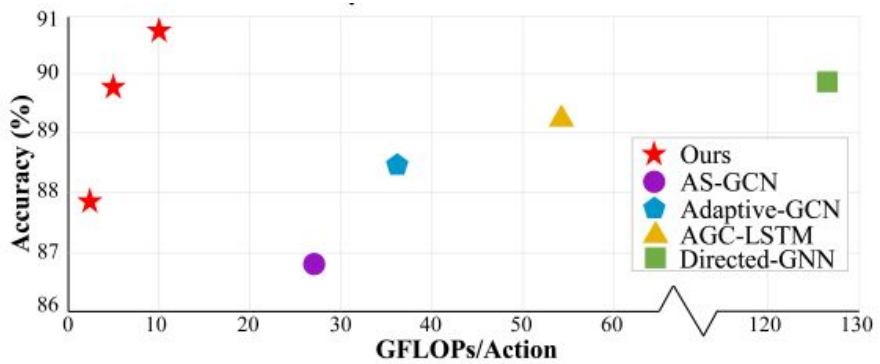

Figure 1. GFLOPs *v.s.* accuracy on NTU RGB+D X-sub task.

Shift convolution in Spatial GCNs

- 그림 2) CNN에서의 Shift와 GCN에서의 Shift 적용 시각화
  - (a) 일반적인 CNN - FLOPS = 커널사이즈^2 * 입력사이즈^2 * 입력채널 * 출력채널
  - (c) 일반적인 GCN - 커널사이즈 처럼 자신/부모/자식 joint에 대한 label을 달리하여 1*1 Conv 연산
  - (b) Shift CNN - 원래 만들려고 했던 커널사이즈 만큼 픽셀 위치를 이동시키는 shift집합을 만들어서 채널별로 서로 다른방향으로 픽셀위치를 옮기고 1*1 Conv로 연산하기
  - (d) Shift GCN - 커널사이즈를 1로하는 대신 인접 노드끼리 입력 feature를 shift하여 1*1Conv 연산 (커널사이즈가 1이므로 - uni-partitioning과 같음)
    - 본 논문은 (d)를 local과 non-local 두 방식으로 shift함

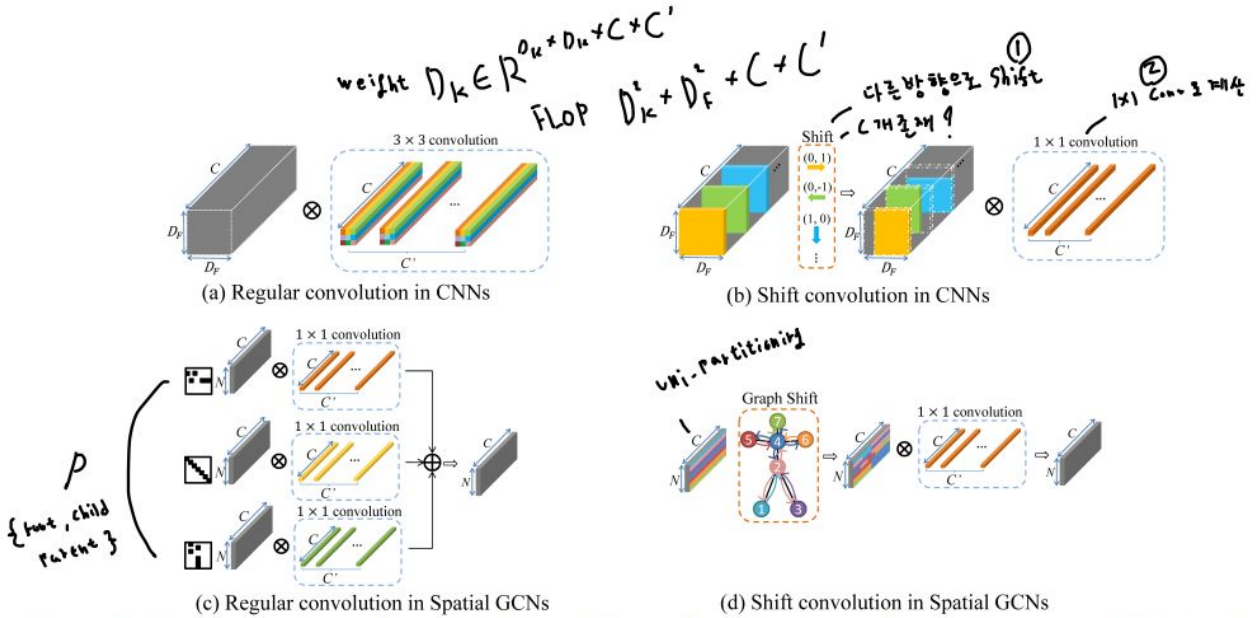

Figure 2. The diagram of regular convolution (a), shift convolution in CNNs (b), and the regular convolution in spatial GCNs (c). Our spatial shift graph convolution is illustrated as (d).

Local Shift Graph Convolution / Non-Local Shift Graph Convolution

- Shift CNN처럼 GCN에도 Shift를 하기 위해 Local Shift Graph Convolution / Non-Local Shift Graph Convolution 제안
- 논문은 모델 설계단계에서 둘중 한가지를 사용하고 실험을 통해 Non-Local이 더 성능향상됨을 보여줌
- Local Shift Graph Convolution => **그림 3의 (a), 식 (3)**
  - 물리적으로 연결되어있는 joint간 입력 feature shift 진행
  - joint 별로 입력 feature size / (인접관절수+1) 를 계산한 뒤 인접 joint의 feature를 concat
  - 이후 1-hop+uni-partitioning+1x1 Conv 로 GCN 연산
  - 문제 1) 일부 joint는 shift 과정에서 자기의 feature정보가 사라지는 문제가 있음
    => **그림 3 (a)**의 3번 joint(보라색)의 뒤에서 5개의 feature가 사라짐
  - 문제 2) 여러 논문에서 그렇듯 멀리떨어진 joint간 관계성 학습이 중요한데 Local Shift는 그러지 않음
- Non-local Shift Graph Convolution => **그림 3의 (b), 식 (4)**
  - Local Shift의 문제점을 보완하기위해 모든 joint가 연결되어있다고 가정하고 Shift
    => **그림 3 (b)** 와 같이 입력 전체 joint 수를 주기로 shift
  - 단, joint 간 관계성이 불필요한 / 중요한 것을 학습하기 위해
    learnable Mask M을 사용 (차원 : joint 수 X 입력 feature 사이즈)
  - Non-Local Shift Graph 결과와 마스크끼리 element-wise product

$\mathbf{F} \in \mathbb{R}^{N \times C}$ $\widetilde{\mathbf{F}} \in \mathbb{R}^{N \times C}$ $B_v = \{B_v^1, B_v^2, \cdots, B_v^n\}$ denote the set of its neighbor nodes, where n denotes the number of neighbor nodes of v.

$$\widetilde{\mathbf{F}}_v = \mathbf{F}_{(v,:c)} \| \mathbf{F}_{(B_v^1,c:2c)} \| \mathbf{F}_{(B_v^2,2c:3c)} \| \cdots \| \mathbf{F}_{(B_v^n,nc:)} \quad (3)$$

$$\widetilde{\mathbf{F}}_M = \widetilde{\mathbf{F}} \circ Mask = \widetilde{\mathbf{F}} \circ (tanh(\mathbf{M}) + 1) \quad (4)$$

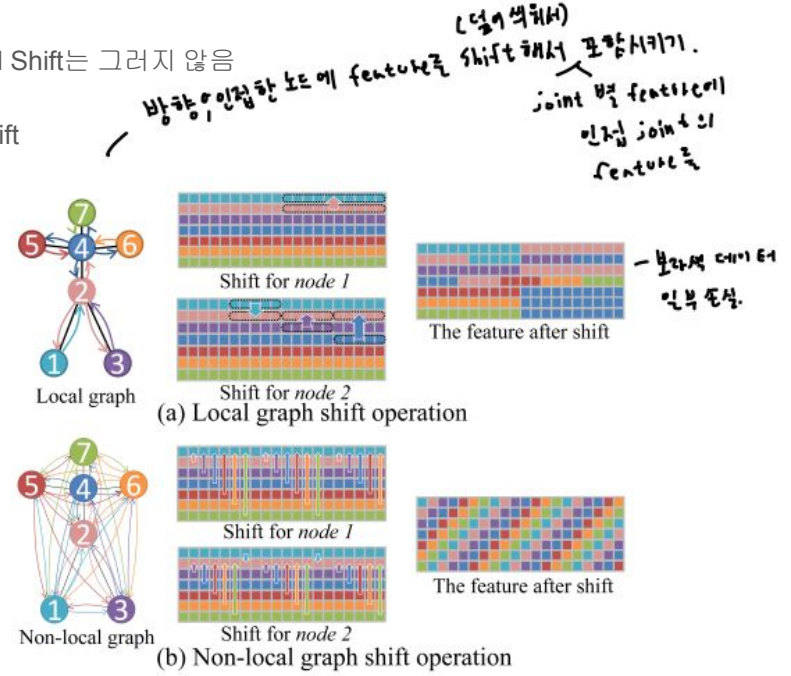

Figure 3. The process of spatial shift graph operation.

Naive temporal shift graph convolution / Adaptive temporal shift graph convolution

- 기존 논문들은 K*1 kernel로 시간축에 대해서 Conv함
  보통 K=9를 사용하였음
- Naive temporal shift graph convolution
  - 커널 사이즈만큼의 shift distance를 만들고 시간축에 대해서 feature별로 shift 시키기
  - 이 기법으로도 기존 TCN과 FLOPS 차이가 9배
  - 단, 여전히 커널 사이즈가 고정되어 생기는 문제
    - 데이터셋마다 프레임 길이가 다른 것에 대해 적응하지 못함
    - 레이어 별로 다른 커널 사이즈가 필요할 수 있는데 고려되지 않음
- Adaptive temporal shift graph convolution => 식 (5)
  - Naive temporal shift graph convolution 의 문제를 개선하기 위해 learnable temporal parameter S_i - input feature 별 shift distance 집합 - 제안
  - S_i가 실수이므로 인접 프레임을 정보를 weighted sum
    ex) S_1=1.5이면 1프레임과 2프레임 정보를 weighted sum
- 그림 4) - 논문이 제안하는 Shift Graph Conv를 활용한 모델 구조
  - 실험을 통해 기존 모델(그림 4-(a))와 성능 비교

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

$$-u, -u+1, \cdots, 0, \cdots, u-1, u$$

$$S_i,\ i = 1, 2, \cdots, C$$

$$\lambda = S_i - \lfloor S_i \rfloor$$

$$\widetilde{\mathbf{F}}_{(v,t,i)} = (1-\lambda) \cdot \mathbf{F}_{(v,\lfloor t+S_i \rfloor, i)} + \lambda \cdot \mathbf{F}_{(v,\lfloor t+S_i \rfloor+1, i)} \quad (5)$$

소수점이면 인접 프레임정보를 weighted sum 적용

feature size

N x T x C

N: joint, T: time, C: channel

feature 별로 temporal shift.

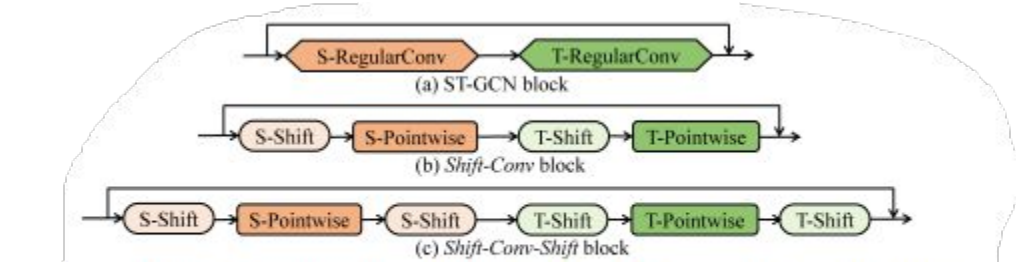

Figure 4. Two modes of combining the shift operation and point-wise convolution.

실험 결과

- NTU RGB+D X-view task 로 ablation study
  - 공간축
    - 표 1) 논문의 공간축에 대한 shift + 모델구조에 따른 성능비교
      Spatial point-wise : ST-GCN
    - 표 2) 기존 논문의 모델과 비교 / FLOPS + Top 1
  - 시간축
    - 표 3) 논문의 시간축에 대한 Shift + 모델구조에 따른 성능비교
      Temporal point-wise : ST-GCN

Model	Shift mode	Top 1
Spatial point-wise	-	90.9
Local shift	Shift+Conv	93.5
	Shift+Conv+Shift	93.9
Non-local shift	Shift+Conv	94.0
	Shift+Conv+Shift	94.2
	Shift+Mask+Conv+Shift	**94.5**

Table 1. Comparisons between the spatial point-wise convolution and our spatial shift graph convolution.

Model	Spatial FLOPs (G)	Top 1
ST-GCN [34]	4.0	93.4
Adaptive GCN [21]	4.0	93.9
Adaptive-NL GCN [21]	5.7	94.2
ST-GCN (one A)	1.3	92.1
Adaptive GCN (one A)	1.3	92.9
Local shift GCN	**1.1**	93.9
Non-local shift GCN	**1.1**	**94.5**

Table 2. Comparisons between regular spatial GCNs and our spatial shift graph GCN.

Model	Shift mode		Top 1
Temporal point-wise	-		79.2
Regular conv(k_t=3)	-		93.4
Regular conv(k_t=5)	-		93.6
Regular conv(k_t=7)	-		**93.7**
Regular conv(k_t=9)	-		93.4
Regular conv(k_t=11)	-		93.4
Naive shift	Shift+Conv	u=1	93.2
		u=2	93.2
		u=3	93.4
		u=4	93.4
	Shift+Conv+Shift	u=1	93.0
		u=2	93.0
		u=3	**93.6**
		u=4	93.3
Adaptive shift	Shift+Conv		94.0
	Shift+Conv+Shift		**94.2**

Table 3. Comparisons between temporal point-wise convolution, regular temporal convolution, naive temporal shift convolution and adaptive temporal shift convolution. The computation cost of temporal shift convolution is $k_t\times$ less than regular temporal convolution, where k_t is the kernel size of regular temporal convolution.

실험 결과

- 그림 5) adaptive temporal shift의 S 시각화
  - (a) NTU RGB+D 는 평균 액션 프레임 길이가 71.4 frame
  - (b) NW-UCLA는 평균 액션 프레임 길이가 39.4 frame
  - Shift-Conv-Shift 모델로 20개의 S_i가 존재
  - 하위 레이어는 공간 관계학습이 중요한지 temporal distance 값들이 낮음
  - ‘Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification.’ 논문에서는 적절한 temporal Conv 위치를 찾기위해 탐색했지만 본논문은 일괄적용
- 표 4) 제안 기법 2개를 조합해서 성능비교

Spatial model	Temporal model	FLOPs (G)	Top 1
Regular S-GCN	Regular T-GCN	16.2	93.4
Shift S-GCN	Regular T-GCN	13.3	94.5
Regular S-GCN	Shift T-GCN	5.4	94.2
Shift S-GCN	Shift T-GCN	**2.5**	**95.1**

Table 4. The effectiveness and efficiency of spatiotemporal shift graph convolution. The accuracy is on NTU RGB+D X-view task.

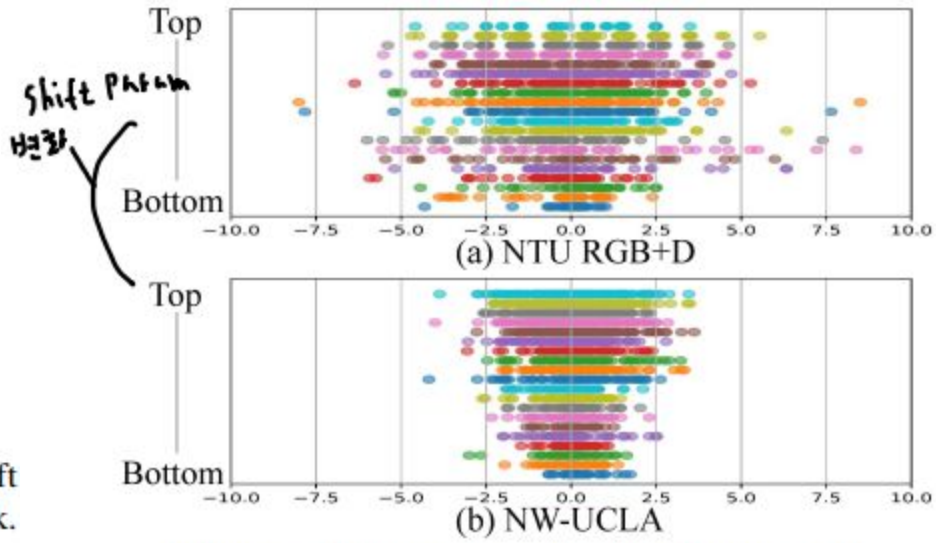

Figure 5. Visualization of adaptive temporal shift.

실험 결과

- 타 논문들과 데이터셋별로 성능비교
  - 사용한 앙상블
    - 1s : Joint 3d position을 입력으로 하는 모델 1개
    - 2s : Joint + Bone(joint끼리 빼서 백터화)을 각각 입력
    - 4s : 2s + 2s의 입력값 다음 프레임간 차이를 입력

Methods	X-view	X-sub	FLOPs (G)
Lie Group [26]	52.8	50.1	-
HBRNN [2]	64.0	59.1	-
Deep-LSTM [19]	67.3	60.7	-
VA-LSTM [35]	87.7	79.2	-
TCN [7]	83.1	74.3	-
Synthesized CNN [18]	87.2	80.0	-
3scale ResNet 152 [10]	90.9	84.6	-
ST-GCN [34]	88.3	81.5	-
Motif+VTDB [31]	90.2	84.2	-
2s AS-GCN [12]	94.2	86.8	27.0
2s Adaptive GCN [21]	95.1	88.5	35.8
2s AGC-LSTM [22]	95.0	89.2	54.4
4s Directed-GNN [20]	96.1	89.9	126.8
1s Shift-GCN (ours)	95.1	87.8	**2.5**
2s Shift-GCN (ours)	96.0	89.7	5.0
4s Shift-GCN (ours)	**96.5**	**90.7**	10.0

① joint, ② bone, ③ joint motion, ④ bone motion

Table 5. Comparisions of the Top-1 accuracy (%) with the state-of-the-art methods on the NTU RGB+D dataset.

Methods	X-sub	X-setup	FLOPs (G)
Part-Aware LSTM [19]	25.5	26.3	-
ST-LSTM [16]	55.7	57.9	-
Multi CNN + RotClips [6]	62.2	61.8	-
SkeMotion [17]	67.7	66.9	-
TSRJI [1]	67.9	62.8	-
1s Shift-GCN (ours)	80.9	83.2	**2.5**
2s Shift-GCN (ours)	85.3	86.6	5.0
4s Shift-GCN (ours)	**85.9**	**87.6**	10.0

Table 7. Comparisions of the Top-1 accuracy (%) with the state-of-the-art methods on the NTU-120 RGB+D dataset.

Methods	Top-1	FLOPs (G)
Lie Group [26]	74.2	-
Actionlet ensemble [28]	76.0	-
HBRNN-L [2]	78.5	-
Ensemble TS-LSTM [8]	89.2	-
2s AGC-LSTM [22]	93.3	10.9
1s Shift-GCN (ours)	92.5	**0.2**
2s Shift-GCN (ours)	94.2	0.3
4s Shift-GCN (ours)	**94.6**	0.7

Table 6. Comparisions of the accuracy (%) with the state-of-the-art methods on the Northwesten-UCLA dataset.