भारतीय प्रौद्योगिकी संस्थान पटना
Indian Institute of Technology Patna

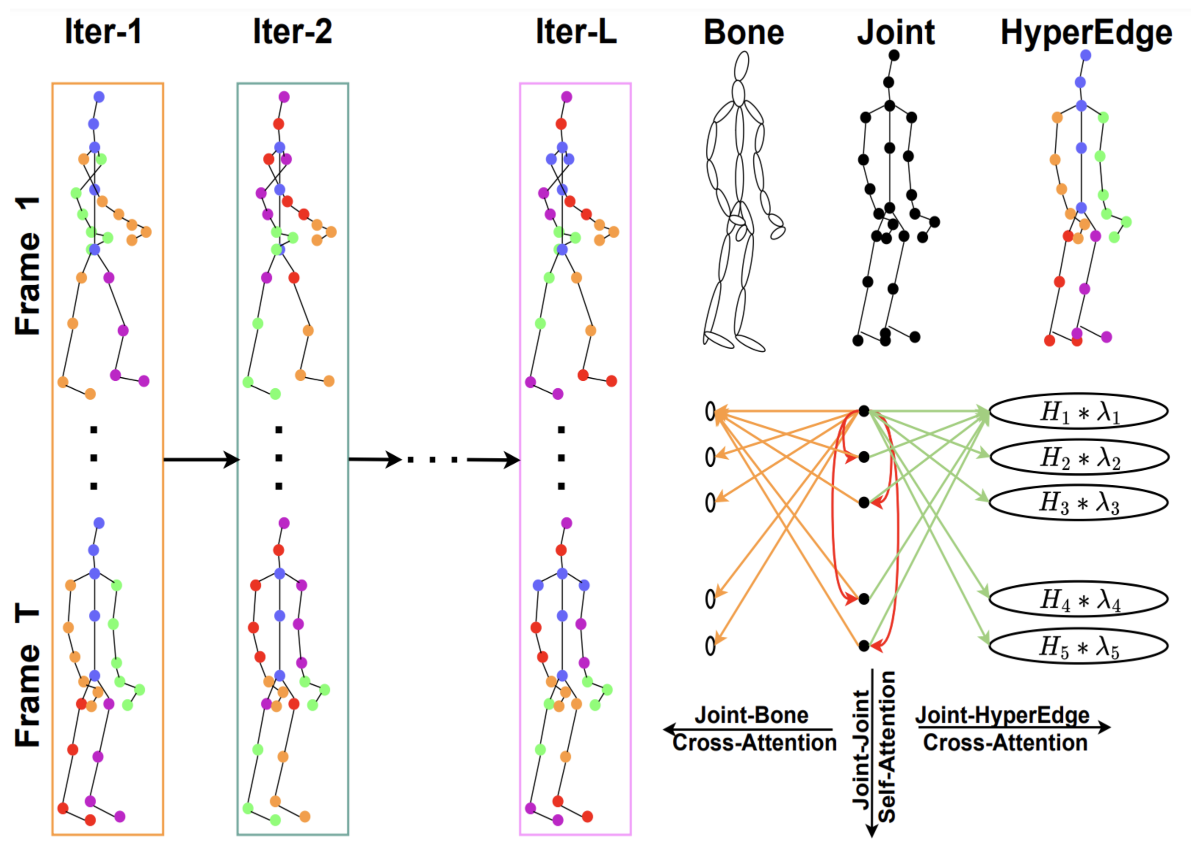
Autoregressive Adaptive Hypergraph Transformer for Skeleton-based Activity Recognition

WACV 2025
TUCSON, ARIZONA • FEB 28 - MAR 4

Abhisek Ray | Ayush Raj | Maheshkumar H. Kolekar

Overview

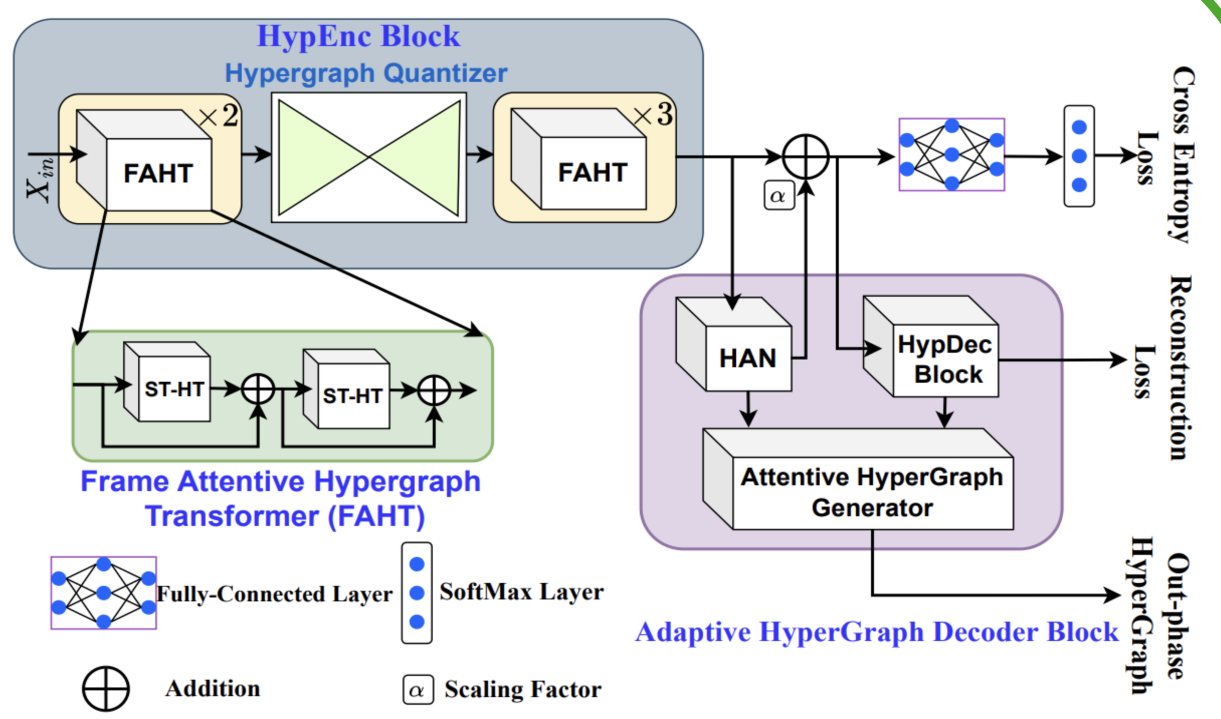
Visual Representation



Motivation

- **Adaptive-** The set of joints to be grouped together in a hyperedge should be action dependent.
- **Autoregressive-** Hypergraph should be learnable just like the parameters of the model

Overall Framework



$H_x = \{m_x^1, m_x^2, \ldots, m_x^{h_x}\}$

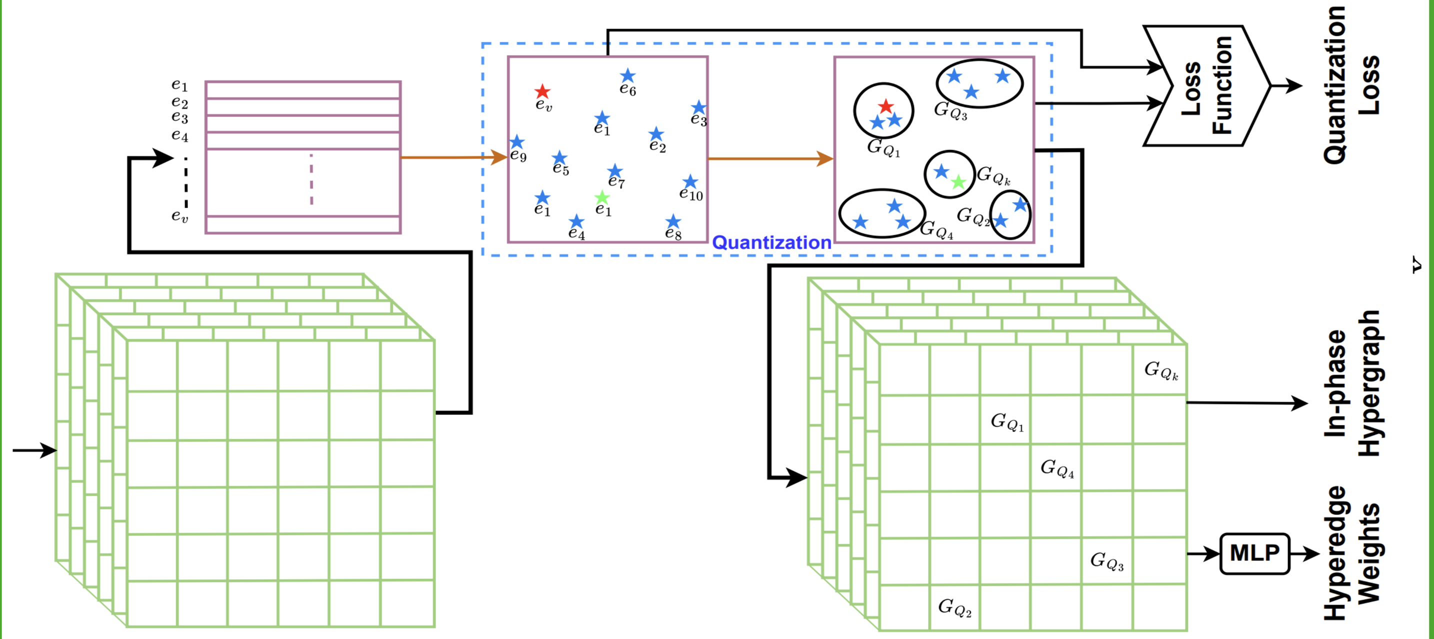
$|H_1| + |H_2| + |H_3| + |H_4| + |H_5| = N$

$\lambda_x = ScalingFactor$

- FAHT block contains hypergraph convolution for extracting features.
- Three kinds of attention operations are used on the extracted features– joint-bone , joint-joint and joint-hyperedge .

Architecture detail

Hypergraph Quantizer

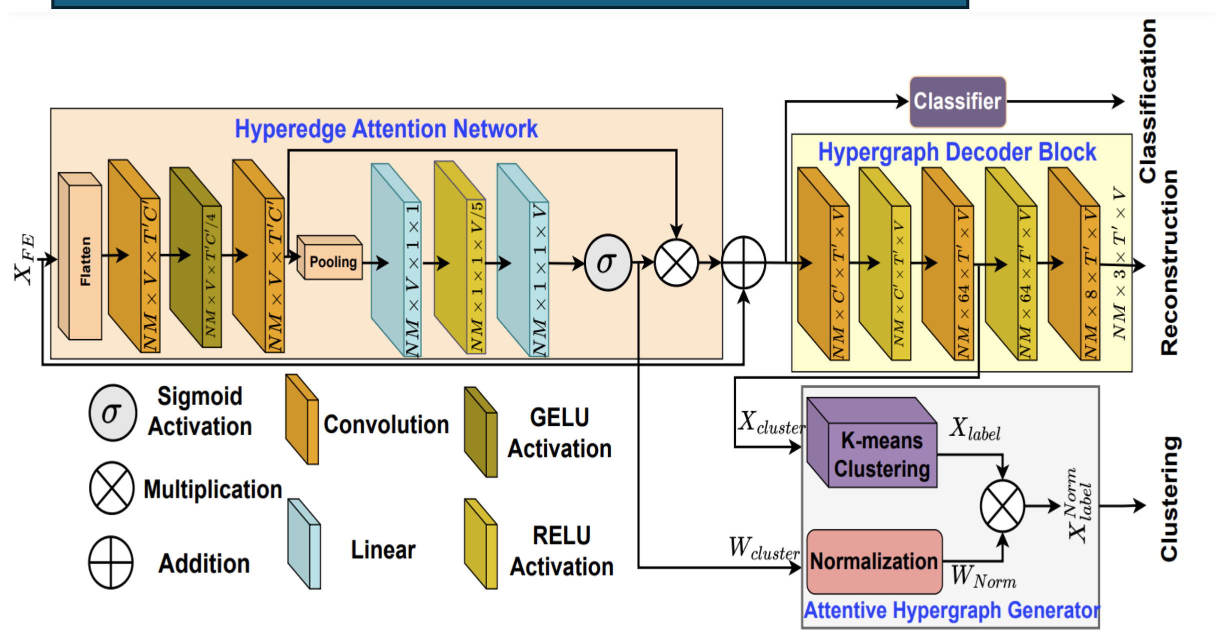


- We have a codebook and the emebedding for each node is quantized into a vector in codebook.

$$k = argmin_j \|E - Q_j\|_2 \qquad \mathcal{L}_{\text{quant}} = (1/V)\sum_{j=1}^{V}\|Q_j - sg(E)\|_2^2.$$

- As the discretizing process is non differentibale, straight through estimator is used for backpropagation.

Adaptive HyperGraph Decoder Block



- Hypergraph Attention Network is responsible for generating the weights of the hyperedges.

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{c=1}^{C}\left\|X_{t,j,c}^{(n)} - \hat{X}_{t,j,c}^{(n)}\right\|^2 \qquad \mathrm{H}_{i,j}^{l+1} = \begin{cases} 1 & \text{if joint } i \text{ is assigned to cluster } j \\ 0 & \text{otherwise} \end{cases}$$

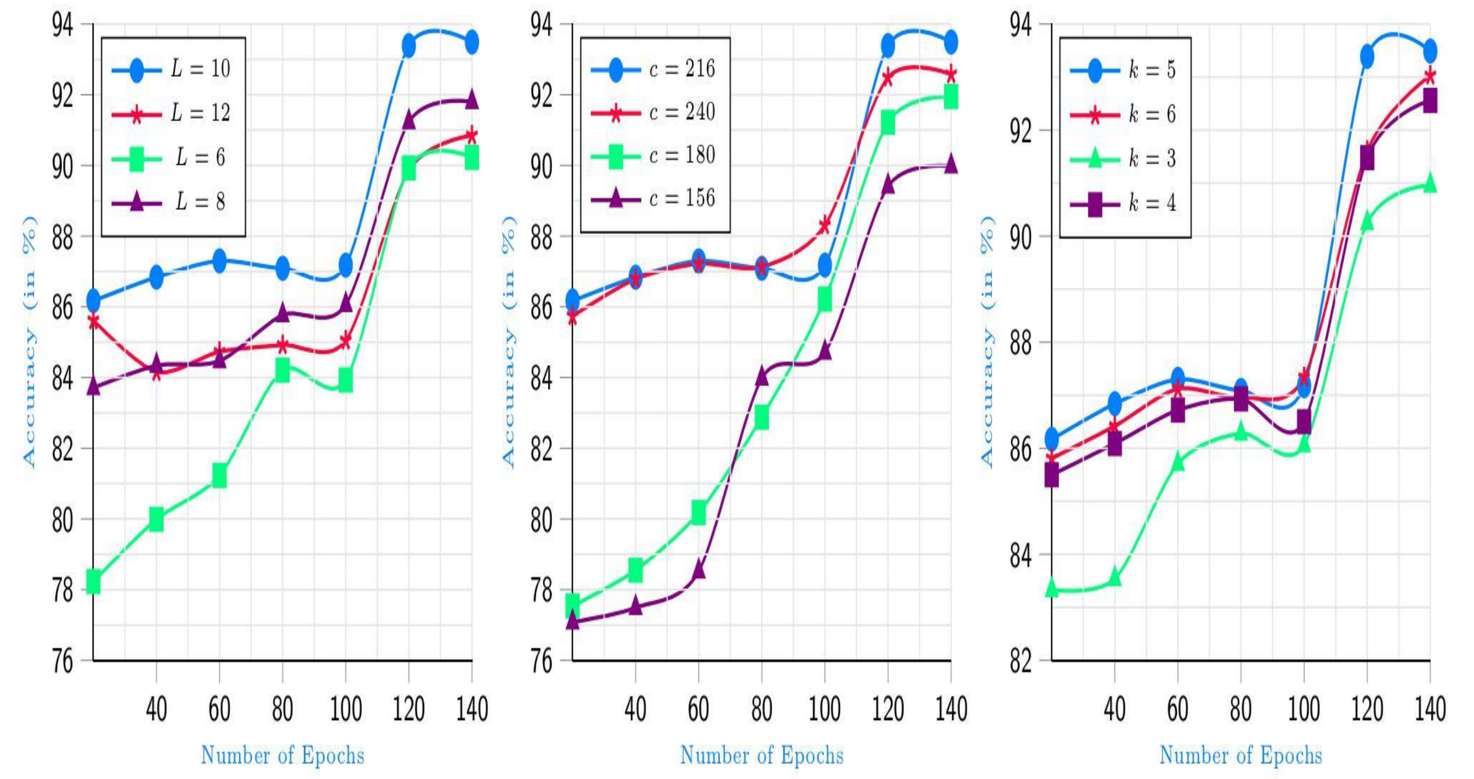
- Reconstruction loss is used as a regularizer for the embeddings extracted from HypEnc block.

Model Algorithm

Algorithm 1 Forward propagation of training process

```
for n ← 1 to total_iteration do
    if n == 1 then
        H^n ← RandomAllocationOfNodesToHyperedges
        H^n_w ← I(Identity Matrix)
    end if
    E_enc ← HypEnc(A, X, H^n, H^n_w)
    A_t, attn ← HAN(E_enc), E_f = E_enc + αA_t
    rec ← HypDec(GAP_time(E_f), A), E_c ← HypDec(GAP_time(E_f), A)
    Feed E_f to classification and reconstruction head
    Update hypergraph for next iteration
    H^{n+1}, H^{n+1}_w ← ATTENTIVE HYPERGRAPH GENERATOR(attn, E_c)
end for
procedure ATTENTIVE HYPERGRAPH GENERATOR(attn, E_c)
    Refer to section 4.2
    return H^{n+1}, H^{n+1}_w
end procedure
```

Ablation



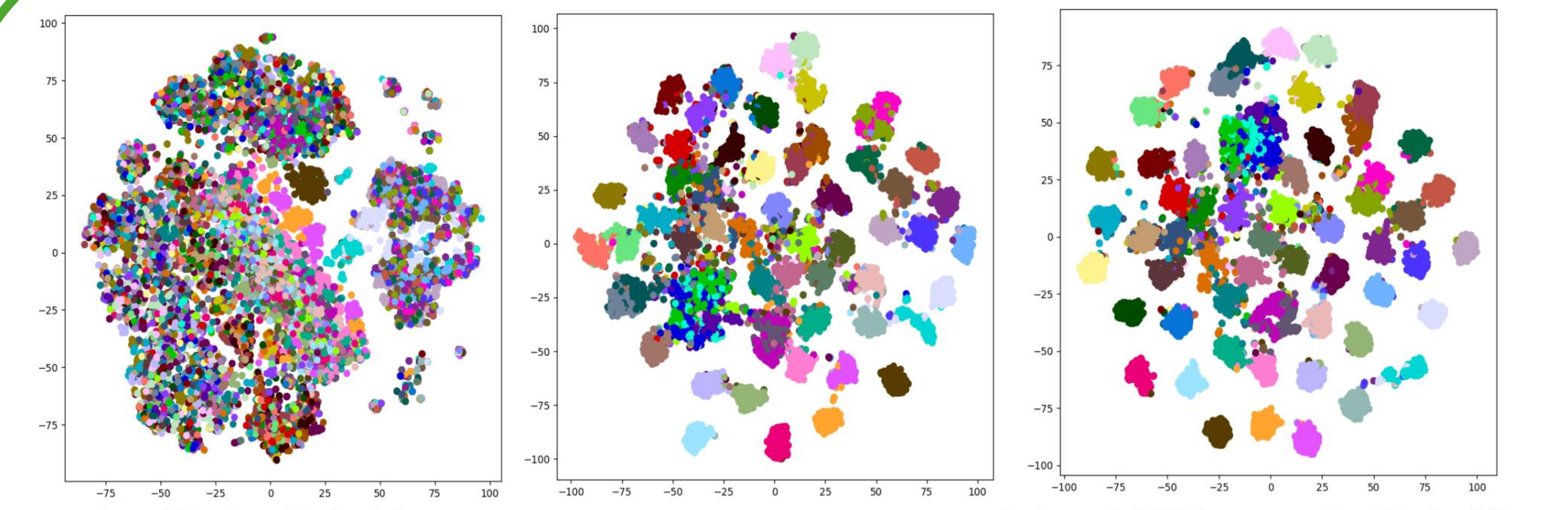
Quantitative Results

Dataset	NTU-60		NTU-120		NW-UCLA
Setting	**X-Sub**	**X-View**	**X-Sub**	**X-View**	
DST-HCN [6]	90.7	96.0	86.0	87.9	-
DST-HCN*	90.7	95.9	85.9	87.9	94.9
DST-HCN*+Ad HypDec	**91.3**	**96.4**	**86.5**	**88.3**	**95.1**
Selective-HCN [9]	90.8	96.6	-	-	-
Selective-HCN*	90.7	96.6	86.3	88.1	95.0
Selective-HCN*+Ad HypDec	**91.4**	**97.0**	**86.7**	**88.3**	**95.3**
Hyperformer [8]	92.9	96.5	89.9	91.3	96.9
Hyperformer*	92.9	96.4	89.9	91.3	96.8
Hyperformer*+Ad HypDec	**93.65**	**97.25**	**90.49**	**91.91**	**97.43**
3Mformer [5]	94.8	98.7	92.0	93.8	97.8
3Mformer*	94.8	98.6	91.9	93.8	97.8
3Mformer*+Ad HypDec	**95.2**	**98.9**	**92.2**	**94.1**	**98.1**

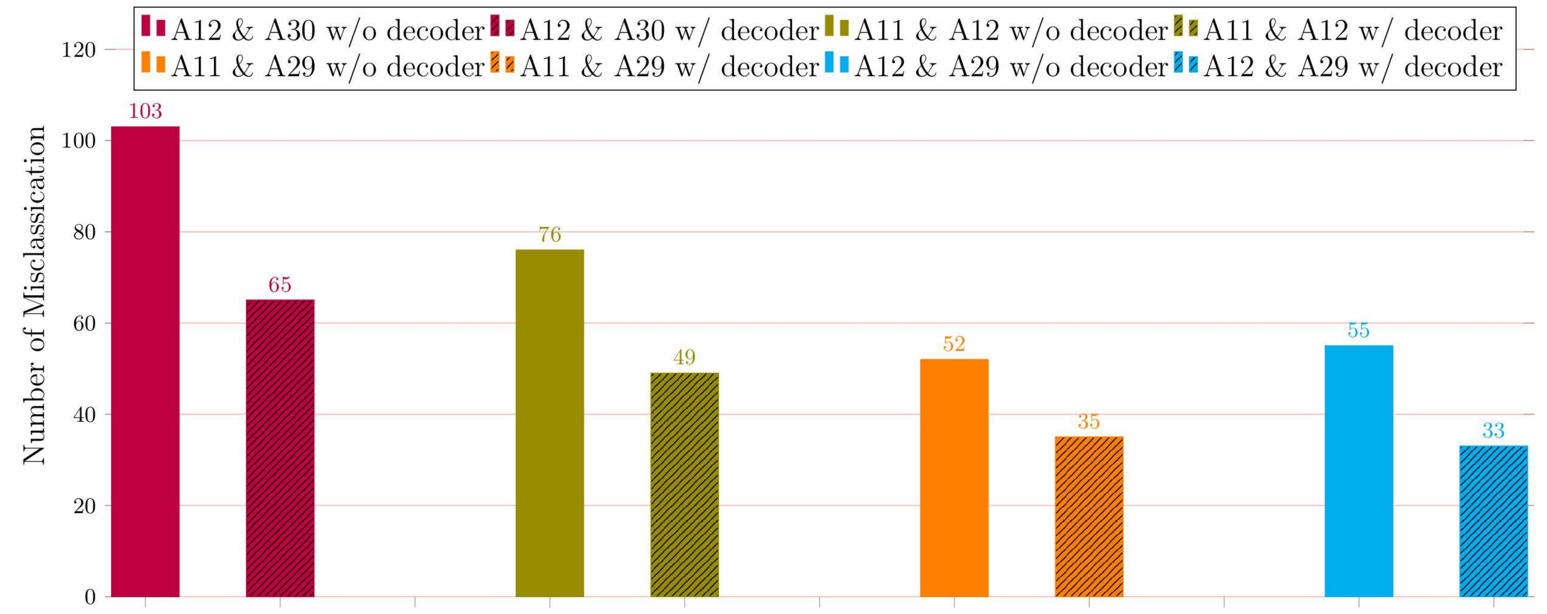
Model Complexity

Models	Publication	Params (M)	Flops (G)	X-Sub/X-View (%)
ST-GCN [7]	AAAI-2018	3.08	16.32	81.5/88.3
Shift-GCN [2]	CVPR-2020	2.76	10.01	90.7/96.5
CTR-GCN [1]	ICCV-2021	5.84	7.88	92.4/96.8
Info-GCN [3]	ICCV-2022	6.28	6.72	92.7/96.9
HD-GCN [4]	ICCV-2023	6.72	6.40	93.0/97.0
DST-HCN [6]	ICME-2023	3.50	2.93	92.3/96.8
Hyperformer [2]	arXiv-2023	2.60	14.8	92.9/96.5
AutoregAd-HGformer	**Proposed**	**3.20**	**15.4**	**94.15/97.83**

Qualitative Results



Input Feature Embedding | Hyperformer Output Embedding | AutoregAd-HGformer output Embedding



References

1. CTR-GCN (Chen et al.), ICCV, pages 13359–13368 2021.
2. Shift-GCN (Cheng et al.), CVPR, pages 183–192, 2020.
3. Info-GCN (Chi et al.), CVPR, pages 20186–20196, 2022.
4. HD-GCN (Lee et al.), ICCV, pages 10444–10453, 2023.
5. 3Mformer (Wang et al.), CVPR, pages 5620–5631, 2023.