Human Pose as Compositional Tokens

Zigang Geng, Chuynyu Wang, Uoxuan Wei, Ze Liu, Houqiang Li, Han Hu
University of Science and Technology of China, Tsinghua University, Microsoft Research Asia 2023

2023. 4. 26.

경영과학연구실　전재현

- Why Human Pose Estimation is Challenging?

No Occlusion

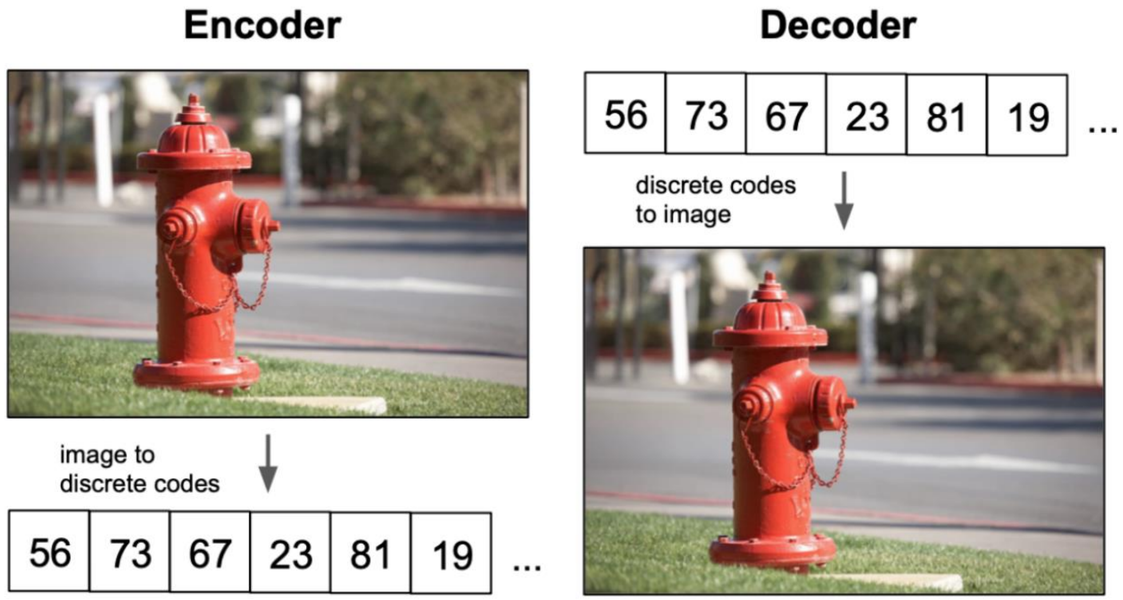
Occlusion

Use context by our experience

But how about computer?

Backgrounds

- VQ-VAE

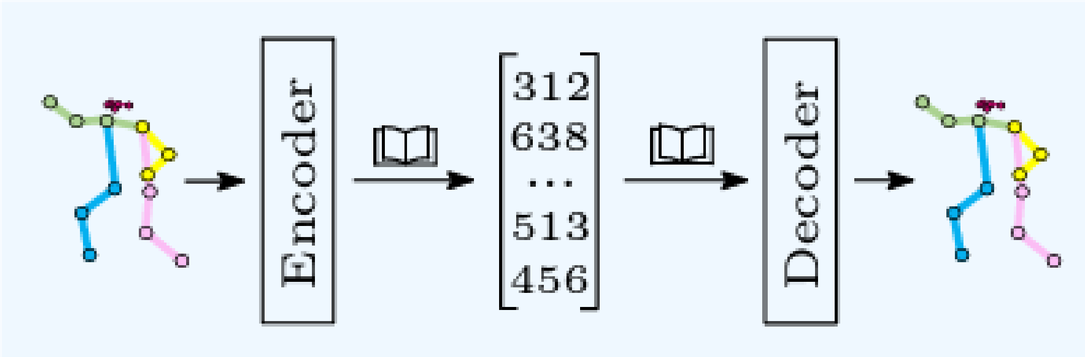


- **Vector Quantized Variational Autoencoder**
- **Consists of two parts : Encoder, Decoder, Codebook**
- **Encoder : Maps an input to a latent code**
- **Decoder : Maps the latent code to a reconstructed image**
- **Codebook : Iteratively updated to best represent the original data**
- **Use sum of different losses : Reconstruction Loss Commitment Loss**

Problem Statement & Key Idea

- Problem Statement
  - **Estimating 2D human pose from a mono-view image or video**

- Key Idea
  - **Representing a pose by discrete tokens rather than heatmaps or coordinates**
  - **Using vector quantizing technique similar with VQ-VAE**
  - **Estimate 2D pose by considering the relationship between joints**



Related works

- Pose representations

  - **Jiefen Li et al. “Crowdpose : Efficient crowded scenes pose estimation and a new benchmark”. In CVPR, 2019**
  - **Zigang Geng et al. “Bottom-up human pose estimation via disentangled keypoint regression”. In CVRP,pages 14676-14686, 2021**
  - **Yuanhao Cai et al. “Learning delicate local representations for multi-person pose estimation”. In ECCV, pages 455-472, 2020.**

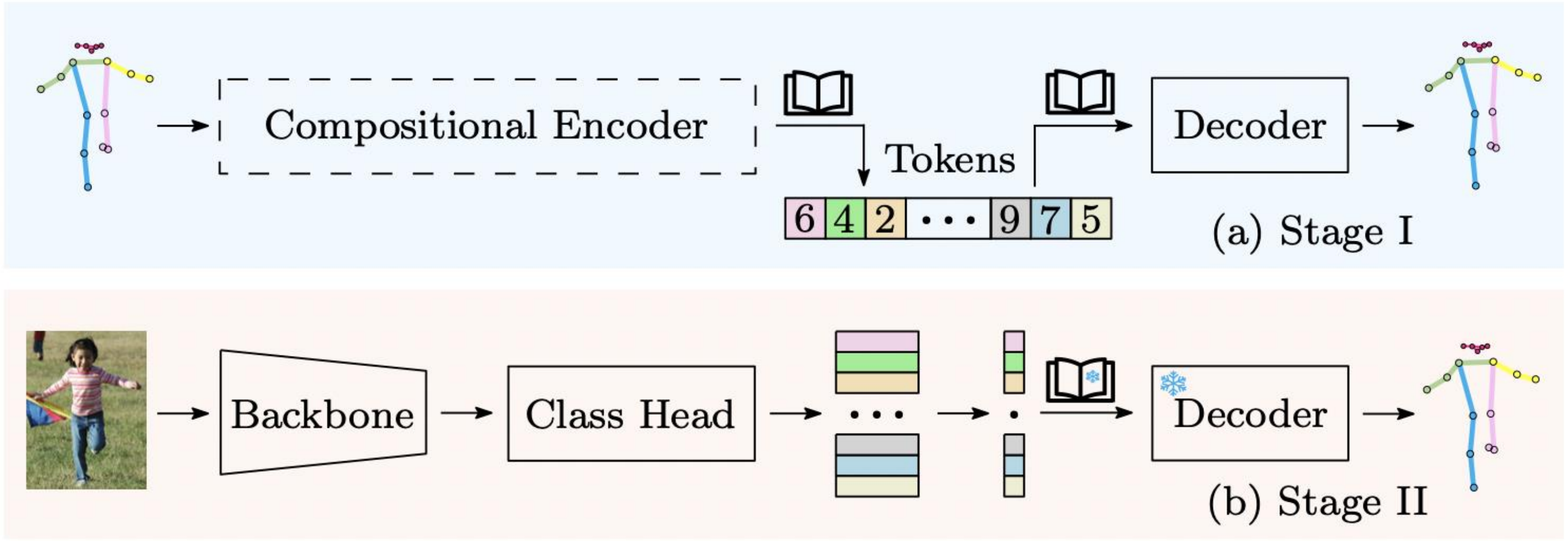
- Modeling joint dependency

  - **Mykhaylo Andriluka et al. “Pictorial structures revisited : People detection and articulated pose estimation”. In 2009 IEEE conference on computer vision and patter recognition, pages 1014-1021. IEEE, 2009**
  - **Zigang Geng et al. “Bottom-up human pose estimation via disentangled keypoint regression”. In CVPR, pages 14676-14686, 2021.**
  - **Jian Wang et al. “Graph-pcnn : Two stage human pose estimation with graph pose refinement”. In ECCV, pages 492-508, 2020.**

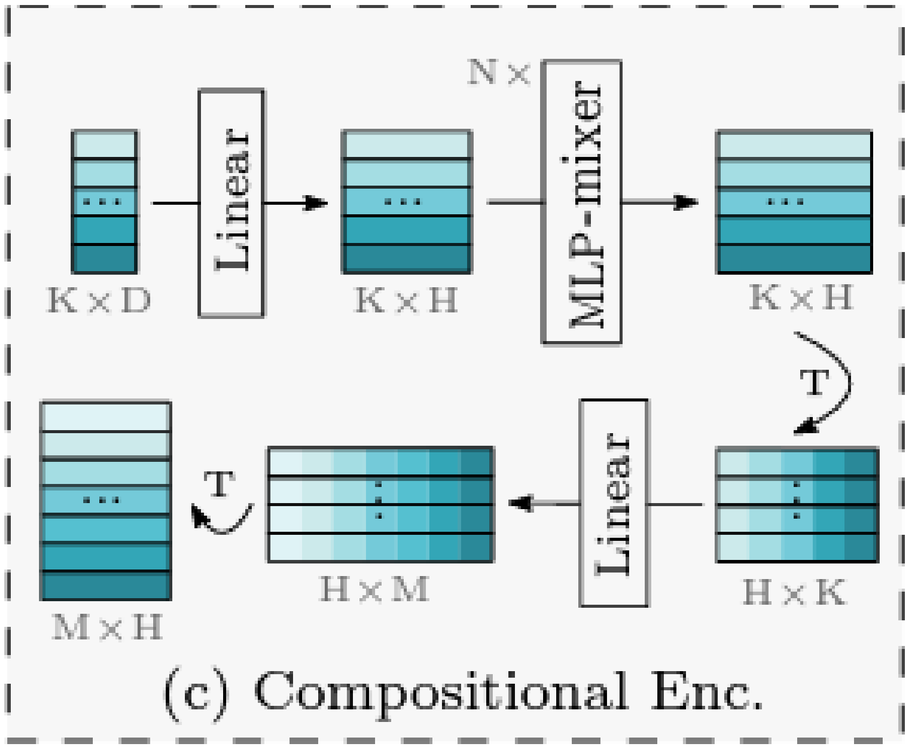
Method

- Overall Structure

  - **Stage 1 : Learning Compositional Encoder, Codebook, Decoder**
  - **Stage 2 : Classification task**
  - **This idea employs the same vector quantizing technique and similar loss function from VQ-VAE**



- Compositional Encoder



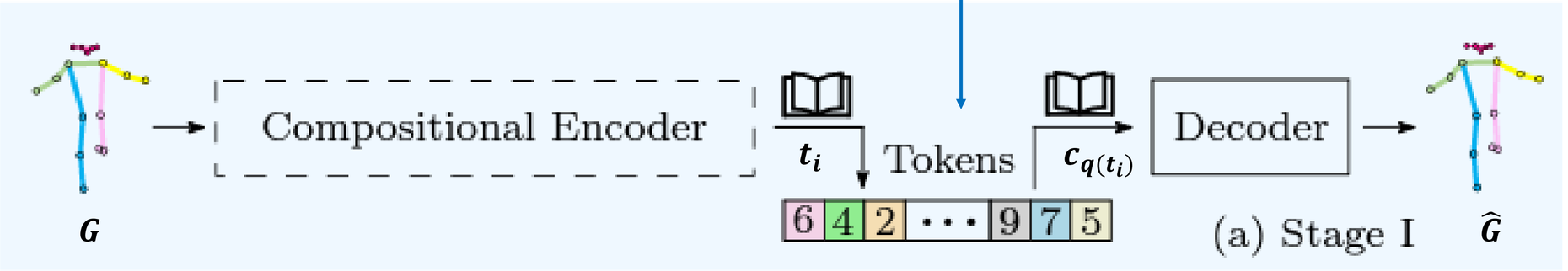
- **Transform a pose into M token features**
- **Input : Raw Pose G(Consist of 2d coordinates of each joints)**
- **Output : Token features(Sub-structure of the pose)**

$$\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_M) = f_e(\mathbf{G})$$

Method

- Stage 1

How to quantize tokens? $q(\mathbf{t}_i = v|\mathbf{G}) = \begin{cases} 1 & \text{if} \quad v = \arg\min_j \|\mathbf{t}_i - \mathbf{c}_j\|_2 \\ 0 & \text{otherwise} \end{cases}$



- **Step 1. Transform a 2d pose into M token features**
- **Step 2. Quantize each token using codebook by the nearest neighbor look-up**
- **Step 3. Transform M tokens into a 2d pose**
- **Encoder network, Codebook, Decoder network**

  → Jointly learned by <u>minimizing following loss function</u>

$$\ell_{pct} = \underbrace{\text{smooth}_{L_1}(\hat{\mathbf{G}}, \mathbf{G})}_{\text{Reconstruction Loss}} + \beta \underbrace{\sum_{i=1}^{M} \|\mathbf{t}_i - \text{sg}[\mathbf{c}_{q(\mathbf{t}_i)}]\|_2^2}_{\text{Commitment Loss}}$$

G : Ground-Truth pose
$\hat{G}$: Output pose from Decoder
sg : stop gradient
t_i : token feature i
$c_{q(t_i)}$: quantized result of t_i

- Stage 1

$$\ell_{pct} = \text{smooth}_{L_1}(\hat{\mathbf{G}}, \mathbf{G}) + \beta \sum_{i=1}^{M} \|\mathbf{t}_i - \text{sg}[\mathbf{c}_{q(\mathbf{t}_i)}]\|_2^2$$

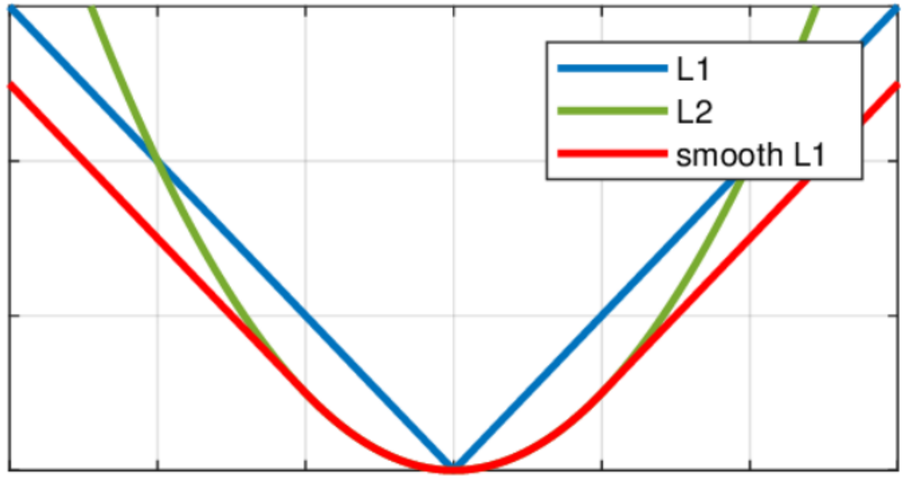
Reconstruction Loss — **Commitment Loss**

G : Ground-Truth pose
$\hat{G}$: Output pose from Decoder
t_i : token feature i
$c_{q(t_i)}$: quantized result of t_i
sg : stop gradient

- **L1 loss : Difference between GT and Predicted value**
- **L2 loss : The squared difference between GT and Predicted value**
- **smooth L1 loss :**

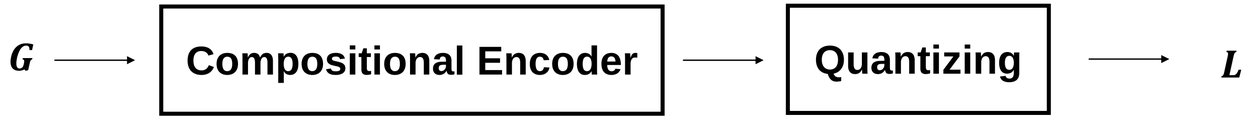
$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

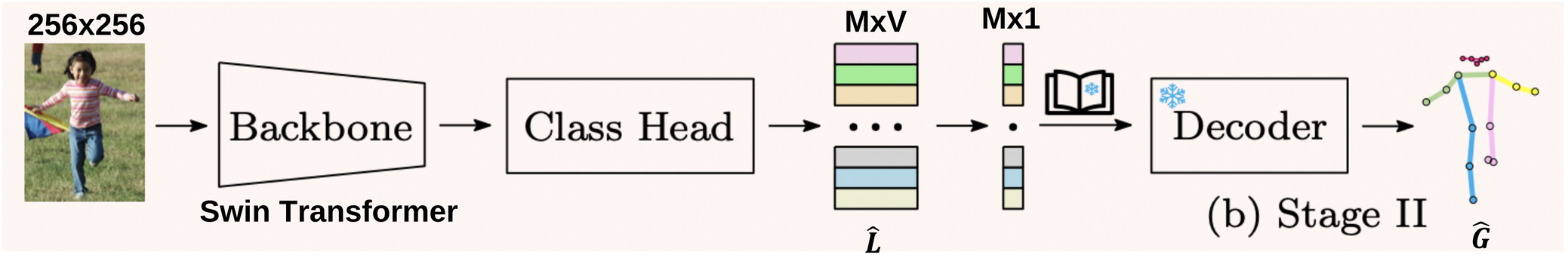
- **stop gradient : Ensuring the codebook is not updated during training encoder**



Method

- Stage 2





- **Classification Head : Predict the categories of the M tokens**
- **Codebook and Decoder are fixed in this stage**
- **Minimizing following Loss function**

$$\ell_{all} = CE(\hat{\mathbf{L}}, \mathbf{L}) + \text{smooth}_{L_1}(\hat{\mathbf{G}}, \mathbf{G})$$

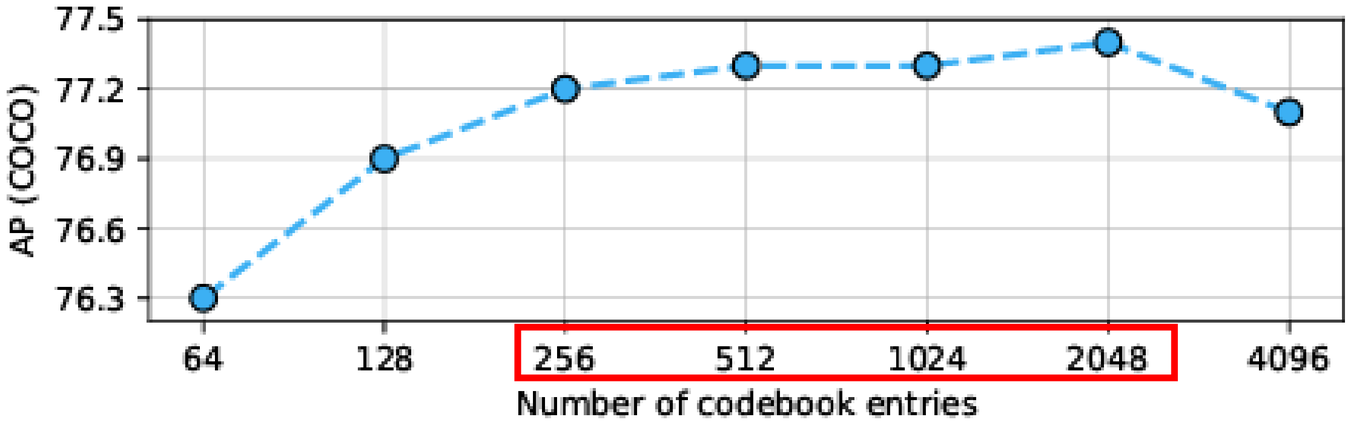
Cross entropy Loss

Difference between Predicted and GT pose

L : Ground-Truth token classes from encoder
$\hat{L}$: Output of classification head

- Experiment Details

  - **Used COCO, MPII(2d), H36M(3d) dataset**
  - **Number of tokens : 34**
  - **Number of codebook entries : 1024**



- Experimental Results

  - **Using Swin-Base performs better than heatmap-based methods(HRNet, HRFormer)**
  - **Using Swin-Huge performs better and faster than ViTPose**

Method	Backbone	Input size	GFLOPs ↓	Speed (fps) ↑	COCO test-dev2017 ↑			COCO val2017 ↑		
					AP	AP^{50}	AP^{75}	AP	AP^{50}	AP^{75}
SimBa. [95]	ResNet-152	384 × 288	28.7	76.3	73.7	91.9	81.1	74.3	89.6	81.1
PRTR [38]	HRNet-W32	384 × 288	21.6	87.0	71.7	90.6	79.6	73.1	89.4	79.8
TransPose [100]	HRNet-W48	256 × 192	21.8	56.7	75.0	92.2	82.3	75.8	90.1	82.1
TokenPose [42]	HRNet-W48	256 × 192	22.1	52.9	75.9	92.3	83.4	75.8	90.3	82.5
HRNet [77, 90]	HRNet-W48	384 × 288	35.5	75.5	75.5	92.7	83.3	76.3	90.8	82.9
DARK [108]	HRNet-W48	384 × 288	35.5	62.1	76.2	92.5	83.6	76.8	90.6	83.2
UDP [31]	HRNet-W48	384 × 288	35.5	67.9	76.5	92.7	84.0	77.8	92.0	84.3
SimCC [41]	HRNet-W48	384 × 288	32.9	71.4	76.0	92.4	83.5	76.9	90.9	83.2
HRFormer [106]	HRFormer-B	384 × 288	29.1	25.2	76.2	92.7	83.8	77.2	91.0	83.6
ViTPose [99]	ViT-Base	256 × 192	17.9	113.5	75.1	92.5	83.1	75.8	90.7	83.2
ViTPose [99]	ViT-Large	256 × 192	59.8	40.5	77.3	93.1	85.3	78.3	91.4	85.2
ViTPose [99]	ViT-Huge	256 × 192	122.9	21.8	78.1	93.3	85.7	79.1	91.6	85.7
SimBa. [95]	Swin-Base	256 × 256	16.6	74.4	75.4	93.0	84.1	76.6	91.4	84.3
Our approach	Swin-Base	256 × 256	15.2	115.1	76.5	92.5	84.7	77.7	91.2	84.7
Our approach	Swin-Large	256 × 256	34.1	76.4	77.4	92.9	85.2	78.3	91.4	85.3
Our approach	Swin-Huge	256 × 256	118.2	31.7	78.3	92.9	85.9	79.3	91.5	85.9

*** AP : Average Precision**

- Experimental Results

  - **Precisions of lower body show large improvement**
  - **Lower body has more occurrences of occlusion than Upper body**

Table 2. Results on the MPII [1] val set (PCKh@0.5).

Method	Hea.	Sho.	Elb.	Wri.	Hip.	Kne.	Ank.	Mean
SimBa. [95]	97.0	95.6	90.0	86.2	89.7	86.9	82.9	90.2
PRTR [38]	97.3	96.0	90.6	84.5	89.7	85.5	79.0	89.5
HRNet [77, 91]	97.1	95.9	90.3	86.4	89.1	87.1	83.3	90.3
DARK [108]	97.2	95.9	91.2	86.7	89.7	86.7	84.0	90.6
TokenPose [42]	97.1	95.9	90.4	86.0	89.3	87.1	82.5	90.2
SimCC [41]	97.2	96.0	90.4	85.6	89.5	85.8	81.8	90.0
Our (Swin-Base)	97.5	97.2	92.8	88.4	92.4	89.6	87.1	92.5

*** PCKh@0.5 : Percentage of Correct Keypoints(threshold : 50% of head bone link**

- Experimental Results

  - **Train the network for 3D poses**
  - **Also shows good performance for 3D pose estimation**

Sharma *et al.* [73]	Zhao *et al.* [112]	Martinez *et al.* [53]	Moon *et al.* [54]	Liu *et al.* [45]	Xu and Takano [98]	Li *et al.* [39]	Gong *et al.* [26]	Zeng *et al.* [107]	*Sun *et al.* [78]	Zou and Tang [114]	*Li *et al.* [36]	Ours (Swin-Base)	Ours (Swin-Huge)
58.0	58.0	57.6	54.4	52.4	51.9	50.9	50.2	49.9	49.6	49.4	48.6	50.8	47.8

*** MPJPE : Mean Per Joint Position Error**

- Conclusion

  - **By using token, the model can incorporate the context of joints**
  - **The model becomes robust to occlusion, by using relationship between joints**
  - **The accuracy of lower body does not exceed 90% yet**