TORE: Token Reduction for Efficient Human Mesh Recovery with Transformer Supplementary Material

Zhiyang Dou[1†] Qingxuan Wu[2†] Cheng Lin[3‡] Zeyu Cao[4‡] Qiangqiang Wu[5]
Weilin Wan[1] Taku Komura[1] Wenping Wang[6]

[1]The University of Hong Kong [2]University of Oxford [3]Tencent Games
[4]University of Cambridge [5]City University of Hong Kong [6]Texas A&M University

This supplementary material covers: network structures and implementation details for both the Encoder-based Transformer (Sec. A1) and Encoder-Decoder-based Transformer (Sec. A2); more comprehensive statistics on model efficiency (Sec. B); visualization of hand vertex-joint interactions (Sec. C); failure cases (Sec. D); visualization of Self-Attention within body joints and vertices (Sec. E); more qualitative comparisons with the state-of-the-art methods (Sec. F); more comparisons with existing token reduction methods (Sec. G) as well as discussion on pruning rate in ITP (Sec. H).

A. Network Structure and Implementation Details

A.1. Transformer Encoder Structure

Pipeline We present the Geometry Token Reduction (GTR) equipped Transformer Encoder Structure based on METRO [8] in Figure A1. The Transformer (Xfmr) Encoder structure is identical to that of METRO [8], with each block comprising a Multi-Head Attention module consisting of 4 layers and 4 attention heads. We employ progressive dimension reduction to decrease the hidden embedding dimensions gradually. However, note that when GTR is utilized, only joint tokens are involved in dimension reduction, and random masking over joint queries is also applied. To query the mesh vertices, NSR uses the learned joint features with 128 feature dimensions produced by the last Xfmr Encoder block and recovers the mesh vertices. For more information on the NSR structure, refer to Sec. A.2. The CNN backbones [4, 16] are initialized with ImageNet-pretrained weight, and extracted image feature size is 2048. The positional encoding is identical to that of [8].

†, ‡ denote equal contributions.

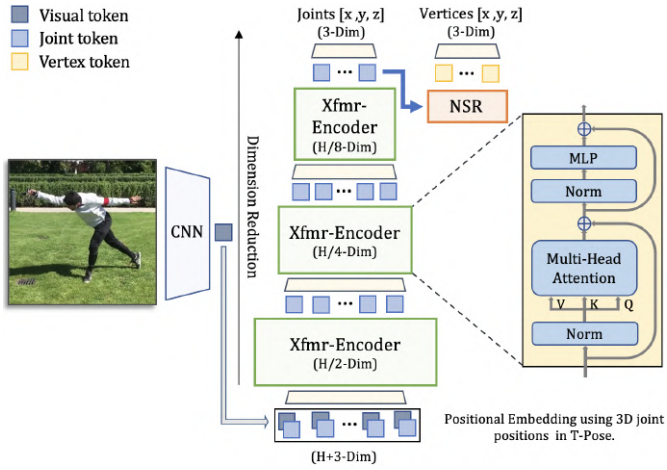

Figure A1. Encoder-based Transformer [8] with Geometry Token Reduction using Neural Shape Regressor. H denotes the dimension of the feature vector.

A.2. Transformer Encoder-Decoder Structure

An overview of the TORE-equipped Transformer Encoder-Decoder structure has been shown in Figure 1 in the main paper. To reduce dimensionality within the Transformer structure, we follow the approach of FastMETRO [3], where we reduce the feed-forward dimension and model dimension from $2048, 512$ to $512, 128$, respectively. The camera token and joint tokens in the Transformer Encoder and Decoder have a dimension size of 512. The Transformer structure of FastMETRO and FastMETRO(S) has 3 and 1 layers, respectively. In the subsequent sections, we elaborate on the Neural Shape Regressor (NSR) and Image Token Pruner (ITP).

Neural Shape Regressor Herein, we present the Neural Shape Regressor (NSR) structure, which is implemented using a Transformer Encoder-Decoder as illustrated in Figure A2. Initially, the joint features $F_J = \{f_1^j, f_2^j, ..., f_J^j\}$, where $f_i^j \in \mathbb{R}^{128\times1}$ and $J = 14$, are processed by a Multi-Head Self-Attention module to improve their representa-

tion. Cross-Attention is then used to learn the interaction between the vertex tokens $T_{\mathcal{V}} = \{t_1^v, t_2^v, ..., t_V^v\}$, where $f_i^j \in \mathbb{R}^{128\times1}$ and $V = 431$, and the learned joint features. Prior to Cross-Attention, Self-Attention is applied among the vertex query tokens, and non-adjacent vertices are masked out to enhance efficiency, as suggested by [3]. The NSR has a feed-forward dimension and model dimension of 512 and 128, respectively, and employs fixed sinusoidal positional encoding [1]. The learned non-local interactions among joints and vertices can be visualized in Sec. E7 and Sec. B4, respectively.

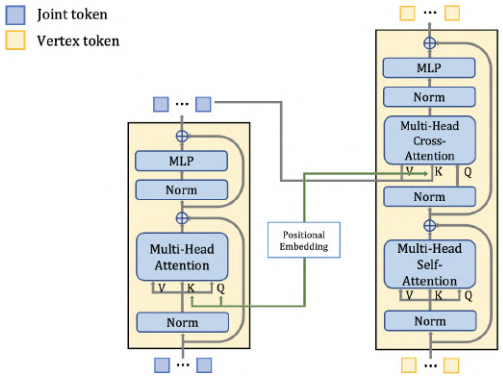

Figure A2. Network of Neural Shape Regressor.

Image Token Pruner Given an input monocular image, the feature is extracted using CNN backbones [4, 16, 14], resulting in a feature map $F_{\mathcal{I}} \in \mathbb{R}^{H\times W\times C}$, where $H = 7$, $W = 7$, and $C = 2048$. After reducing the dimensionality of $F_{\mathcal{I}}$ from C to $C' = 512$, we obtain a dimension-reduced feature map $F'_{\mathcal{I}} \in \mathbb{R}^{H\times W\times C'}$. Subsequently, we flatten $F'_{\mathcal{I}}$ to $F'_{\mathcal{I}} \in \mathbb{R}^{HW\times C'}$ to create HW tokens, which are passed to our ITP module to reduce computational costs in the transformer model.

Training Details TORE-equipped FastMETRO [3] utilizes the same loss terms as described in Sec. A.1. The AdamW optimizer is applied with a learning rate and weight decay of 10^{-4}. Gradient clipping is implemented with a maximal gradient norm value of 0.3. As with METRO [8], the weights of the CNN backbones [4, 16] are initialized using ImageNet-pretrained weights. The low and high-resolution meshes used for hand mesh recovery consist of 195 and 778 vertices, respectively.

B. More Statistics on Model Efficiency

B.1. Model Efficiency at Full Memory Usage

In order to thoroughly examine the capabilities of our models, we perform a throughput analysis using the maximum batch size possible. Our approach involves attempting to fit the largest possible batches into the VRAM of a GPU, specifically an RTX3090 card with 24G VRAM, and conducting a throughput analysis based on this setting. This enables us to uncover the full potential of the models on a consumer-grade graphics card.

Table B1. Comparison for throughput on maximum batch size for monocular 3D human mesh recovery on Human3.6M [6]. We test with ResNet-50 [4] and HRNet-W64 [16] as backbones.

Method	Max BS	GFLOPs ↓	Throughput ↑
METRO-H64	16	56.5	141
METRO-H64+GTR	32	30.3	246.6
FastMETRO-H64	800	35.7	249.7
FastMETRO-H64+GTR+ITP@20%	1248	**30.2**	**302.3**
METRO-R50	24	31.6	247
METRO-R50+GTR	80	5.4	982.5
FastMETRO-R50	1024	10.9	634.2
FastMETRO-R50+GTR+ITP@20%	1120	**5.3**	**1086.4**

As shown in Table B1 demonstrates that using TORE results in enhanced model effectiveness and increased inference throughput. Our approach enables larger batch sizes, higher computational throughput, and lower GFLOPs in computation when the GPU capabilities are maximized, owing to fewer tokens. This renders our approach more practically useful than the prior approach [8, 3].

B.2. Training Memory Cost Comparison

We conducted experiments on the training cost per GPU VRAM to demonstrate our superiority in training resource efficiency. As presented in Table B2 and Table B3, the models that integrate our proposed TOken REduction (TORE) methods display significantly reduced GPU VRAM consumption. Specifically, with TORE, the Transformer Encoder structures [8] with ResNet-50 (R50) [4] and HRNet-W64 (H64) [16] as backbones demonstrate memory savings of 58.3% and 44.2%, respectively. Similarly, the Transformer Encoder-Decoder structures [3] with TORE using ResNet-50 and HRNet-W64 as backbones exhibit reduced memory costs of 27.4% and 17.9%, respectively.

Table B2. Comparison on Training GPU VRAM Cost of the Transformer Encoder structure [8]. The model is trained on 8 GPU cards with a batch size to be 32.

Model	GPU Memory Cost
METRO-H64	32.8GB
METRO-H64+GTR	18.3GB **(-44.2%)**
METRO-R50	24.7GB
METRO-R50+GTR	10.3GB **(-58.3%)**

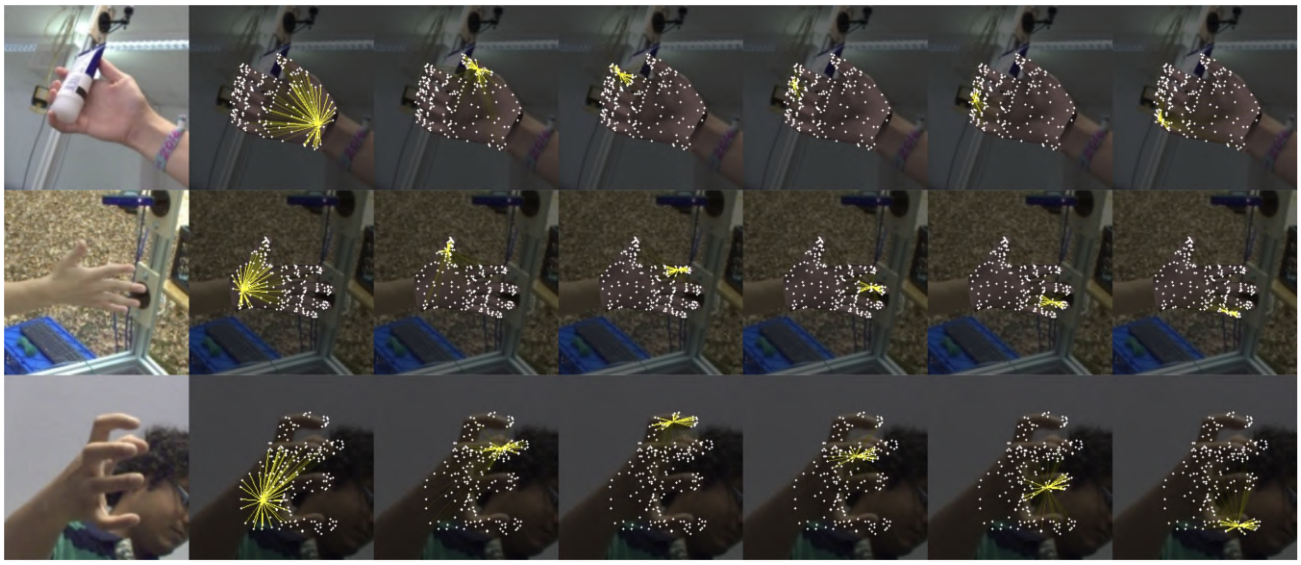

Figure B3. Cross-Attention among hand joints and hand vertices.

Table B3. Comparison on Training GPU VRAM Cost of the Transformer Encoder-Decoder structure [3]. The model is trained on 4 GPU cards with a batch size to be 16.

Model	GPU Memory Cost ↓
FastMETRO-H64	13.4GB
FastMETRO-H64+GTR+ITP@20%	11.0GB **(-17.9%)**
FastMETRO-R50	8.4GB
FastMETRO-R50+GTR+ITP@20%	6.1GB **(-27.4%)**

B.3. Run-time Performance

The run-time performance statistics of our models are presented in Table C4, under the same hardware configuration as Sec. 4.3 in the Main paper.

Notably, we evaluated system performance using established metrics such as throughput and GFLOPs, as utilized in PPT [10], DynamicViT [11], TokenLearner [13], Evit [7], and CrossVit [2], among others. For system performance investigation, Unlike the FPS, which considers the processing of a single instance, throughput is typically used since it measures the maximum number of input instances that the network can process in a given time unit, evaluating the parallel processing of multiple instances [10].

We further discuss the following instance:

Top-Down Human Mesh Recovery During the inference process of the top-down approach in multi-person HMR, an object detector locates multiple human instances in a given input image, which are typically cropped, resized, and grouped into a minibatch for faster inference. The resulting minibatch is then fed into the pose detector. In this common case, we consider throughput a more appropriate metric for evaluating the performance of top-down HMR tasks.

Multi-Camera System In practical use of a multi-camera system, e.g., surveillance, sports analysis, and crowd management, multiple camera feeds are sent to a centralized server for analysis. To efficiently process the aggregated images from multiple cameras, high throughput is required. A high throughput ensures that the system can simultaneously process the camera feeds without any lag or delay in these scenarios. For those offline applications where real-time performance is not highly demanded, a high throughput system also saves time and cost.

C. Vertex-Joint Interactions in Hand Mesh Recovery

Figure B3 demonstrates the interactions modeled by the Neural Shape Regressor (NSR) on Hand Mesh Recovery. To obtain attention scores, we average the scores across all heads of the multi-head cross-attention between query vertices and joint features. The interactions between mesh vertices and joints in the hand model exhibit a shape-blending style similar to MANO [12]. This observation aligns with the human body model and validates the effectiveness of our proposed methods. We used the FastMETRO-H64+GTR+ITP@20% model for visualization.

Table C4. Running time Performance (FPS).

Method	FPS
METRO-H64+GTR	22
FastMETRO-H64+GTR+ITP@80%	26
METRO-R50+GTR	53
FastMETRO-R50+GTR+ITP@80%	61

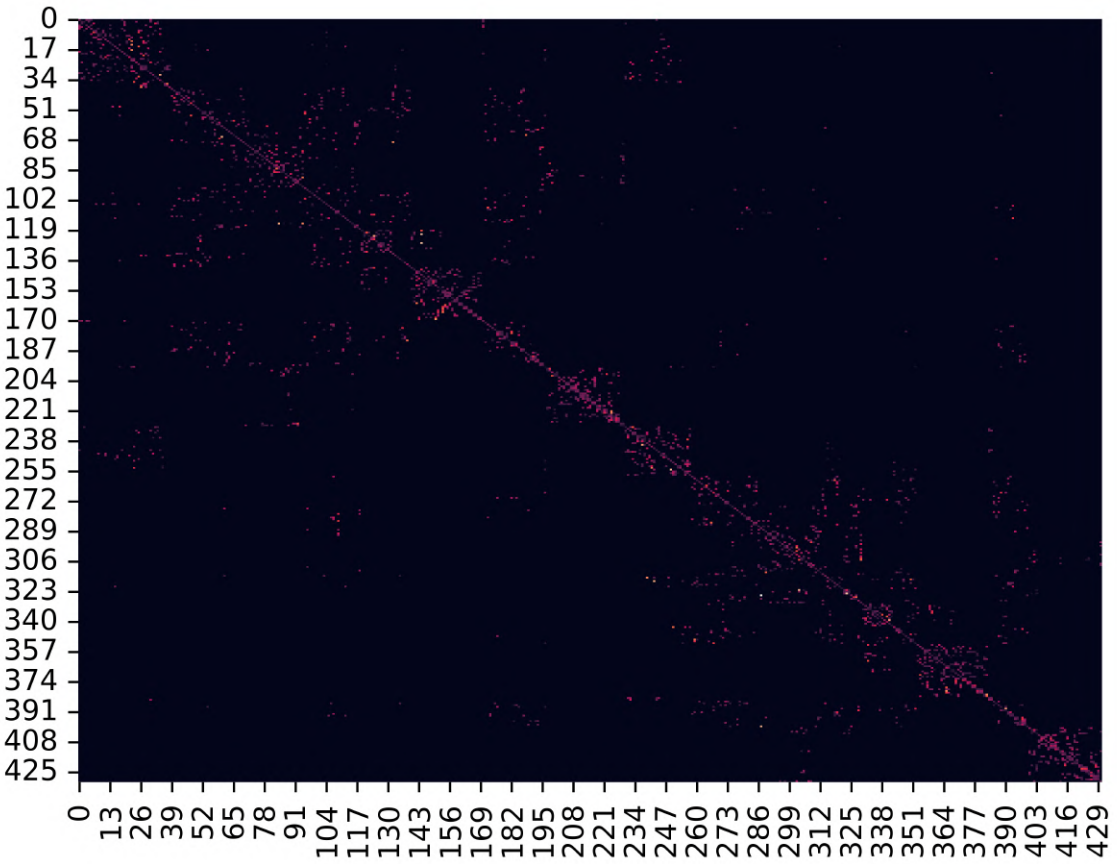

Figure B4. Visualization of Self-Attention within the body vertices.

(a) (b)

Figure C5. Failure cases when the inputs are outside of the training data distribution.

D. Failure Cases

Since Geometry Token Reduction recovers human mesh hierarchically, the quality of the recovered vertices by NSR depends on the learned body features. We conducted two experiments by adding Gaussian noise $\epsilon \in \mathbb{R}^K$ (at $5\%, 7.5\%, 10\%$) to NSR to the body features for mesh vertex regression. K is the dimension of input features. As shown in Figure D6, when the body features are unreliable, e.g., 10% noise level, the performance of human mesh recovery by GTR drops.

Additionally, when the input image is outside of the training data distribution, such as a photo of an infant's

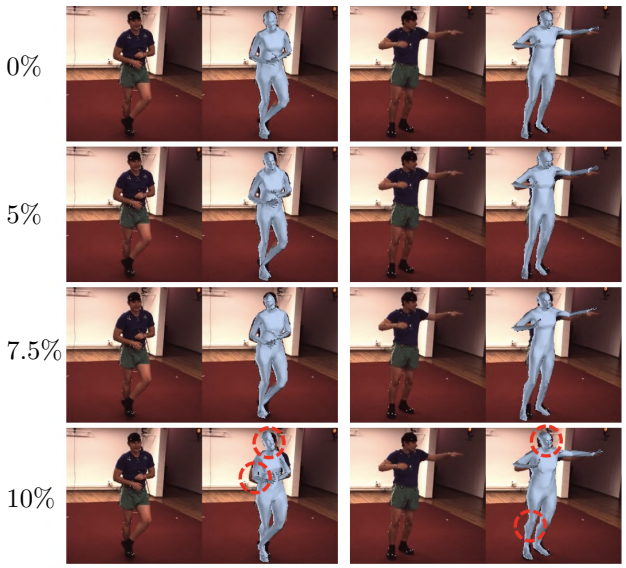

Figure D6. Human mesh recovery results at different noise levels to NSR during GTR. We showcase human mesh recovery at four noise levels: 0%, 5%, 7.5% and 10%.

hand, the recovered mesh's quality may diminish, as shown in Figure C5 (a). Moreover, when the model encounters extremely challenging partial observation, such as an image capturing only one thumb, it cannot recover the hand mesh accurately; see Figure C5 (b). The model used for this study was FastMETRO-H64+GTR+ITP@20%.

E. Self-Attention within Joints and Vertices

In this section, we investigate the interactions between joints and vertices within the Self-Attention module. Specifically, joint interactions are analyzed by averaging attention scores from all Multi-Head Self-Attention modules within the Transformer Encoder, while vertex interactions are assessed by averaging scores from all heads of Multi-Head Self-Attention within the Transformer Decoder. The resulting Self-Attention visualizations for joints and vertices are presented in Figure E7 and Figure B4, respectively. As depicted in Figure E7, Self-Attention effectively models non-local interactions among joints, thus improving model robustness to partial observation and self-occlusion in challenging monocular observations. Similarly, Figure B4 shows that non-local interactions among vertices enhance mesh recovery performance when learned joint features are employed. Visualization was performed using FastMETRO-H64+GTR+ITP@20%.

F. More Qualitative Comparisons

The introduction of TORE greatly saves the computational cost, i.e., GFLOPs and improves the throughput while enabling the model to produce competitive or even better mesh recovery from monocular images. We further conducted qualitative comparisons with existing methods such as [8, 9, 3] for human mesh recovery on 3DPW [15] and Human3.6M [6]. Our results are summarized in Figure H8 and Figure H9, respectively. All methods utilized HRNet-W64 [16] as the CNN backbone, and our model setting is FastMETRO-H64+GTR+ITP@20%.

We also conduct qualitative comparisons with existing methods [5, 9, 3] for human mesh recovery on 3DPW [15] and Human3.6M [6], which are summarized in Figure H8 and Figure H9, respectively. All methods use the HRNet-W64 [16] as a CNN backbone, and our model setting is FastMETRO-H64+GTR+ITP@20%.

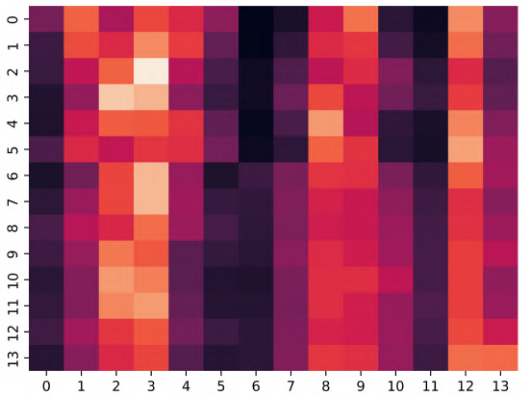

Figure E7. Visualization of Self-Attention within the body joints

Table G5. TCFormer [17] v.s. TORE (Ours) for HMR on Human3.6M.

Method	Throughput↑	3DPW		Human3.6M	
		MPJPE↓	PAMPJPE↓	MPJPE↓	PAMPJPE↓
TCFormer [17]	230.9	80.6	49.3	62.9	42.8
METRO+TORE (Ours)	210.1	75.5	46.6	**57.6**	37.1
FastMETRO+TORE (Ours)	249.2	**72.3**	**44.4**	59.6	**36.4**

G. More Comparisons with Existing Methods

G.1. Comparison with TCFormer [17].

We conduct a comparison between TORE and another token clustering method for HMR. Compared with TCFormer [17], we have *1) Different architectures*. TCFormer is a much more complicated multi-stage method for token clustering, while ours only requires a single pass; *2) Different body representation*. There is no consideration of body representation in TCFormer while we propose NSR for GTR to reduce redundancy; *3) Different performance.* Compared with TCFormer with the same setting in Tab G5, where our method surpasses TCFormer on both two datasets.

G.2. Comparison with PPT [10]

We compared with PPT [10] that prunes tokens by locating human visual tokens according to attention score; see Tab G6 where ours is more competitive.

Table G6. PPT [10] v.s. TORE (Ours). We test with FastMETRO-Eb0 on Human3.6M.

Method	GFLOPs↓	Throughput↑	MPJPE↓	PAMPJPE↓
PPT	1.6	862.1	68.4	46.2
Ours	1.6	**870.4**	**63.2**	**43.9**

H. Influence of Pruning Rate in ITP

The influence of different pruning rates is shown in Tab H7. In this paper, we empirically set the pruning rate to 20% based on extensive experiments.

Table H7. Influence of pruning rates. We test with FastMETRO-Eb0 on Human3.6M.

Pruning rate	PAMPJPE ↓	MPJPE ↓	GFLOPS ↓
No Pruning	45.8	69.2	7.1
0.2	**43.9**	**63.2**	1.6
0.5	44.0	64.2	1.4
0.75	44.7	65.3	**1.2**

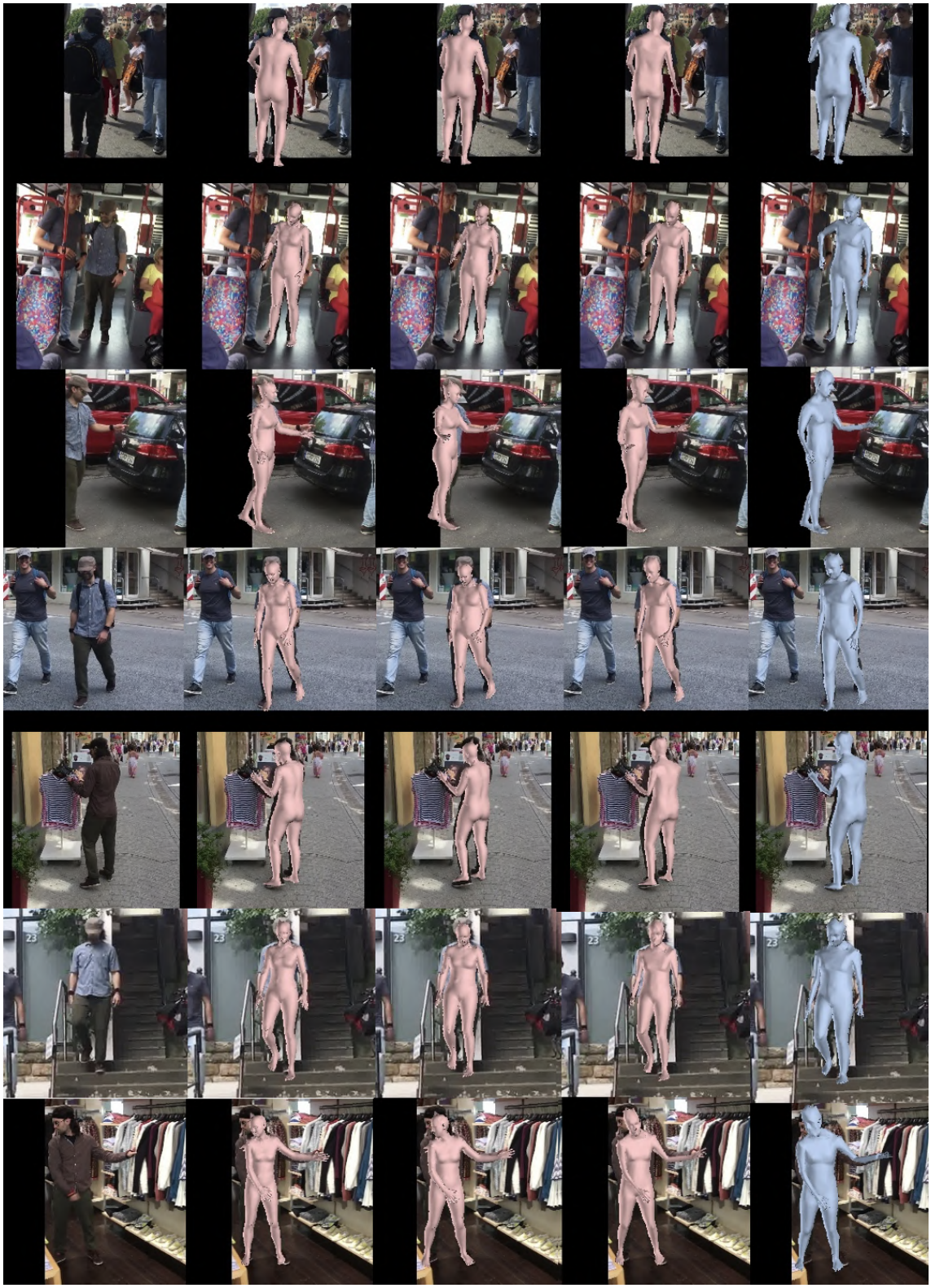

Figure H8. Qualitative Comparison with existing Transformer-based methods [8, 9, 3] on 3DPW [15].

Figure H9. Qualitative Comparison with existing Transformer-based methods [8, 9, 3] on Human3.6M [6].

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[2] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

[3] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *European Conference on Computer Vision (ECCV)*, 2022.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[6] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[7] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

[8] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021.

[9] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12939–12948, 2021.

[10] Haoyu Ma, Zhe Wang, Yifei Chen, Deying Kong, Liangjian Chen, Xingwei Liu, Xiangyi Yan, Hao Tang, and Xiaohui Xie. Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In *ECCV*, 2022.

[11] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

[12] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[13] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. 2021.

[14] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[15] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

[16] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[17] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111, 2022.