# Supplementary Material
# End-to-End Human Pose and Mesh Reconstruction with Transformers

Kevin Lin  Lijuan Wang  Zicheng Liu
Microsoft
{keli, lijuanw, zliu}@microsoft.com

| Method | 3DPW | | | Human3.6M | |
|---|---|---|---|---|---|
| | MPVE ↓ | MPJPE ↓ | PA-MPJPE ↓ | MPJPE ↓ | PA-MPJPE ↓ |
| HMR [11] | – | – | 81.3 | 88.0 | 56.8 |
| GraphCMR [14] | – | – | 70.2 | – | 50.1 |
| SPIN [13] | 116.4 | – | 59.2 | – | 41.1 |
| Pose2Mesh [3] | – | 89.2 | 58.9 | 64.9 | 47.0 |
| I2LMeshNet [15] | – | 93.2 | 57.7 | 55.7 | 41.1 |
| VIBE [12] | 99.1 | 82.0 | 51.9 | 65.6 | 41.4 |
| HoloPose [7] | – | – | – | 60.2 | 46.5 |
| Arnab *et al.* [2] | – | – | 72.2 | 77.8 | 54.3 |
| DenseRaC [17] | – | – | – | – | 48.0 |
| Zhang *et al.* [19] | – | – | 72.2 | – | 41.7 |
| Zeng *et al.* [18] | – | – | – | 60.6 | 39.3 |
| HKMR [5] | – | – | – | 59.6 | 43.2 |
| METRO (Ours) | **88.2** | **77.1** | **47.9** | **54.0** | **36.7** |

Figure 1: Adding references (HKMR [5], Zeng *et al.* [18], Zhang *et al.* [19], DenseRaC [17], Arnab *et al.* [2], HoloPose [7]) to the comparisons on 3DPW and Human 3.6M datasets.

## A. Additional Reference

We would like to add additional references (HKMR [5], Arnab *et al.* [2], Zeng *et al.* [18], Zhang *et al.* [19], DenseRaC [17], HoloPose [7]). Among the new references, HKMR [5] regresses SMPL parameters by leveraging a pre-specified hierarchical kinematic structure that consists of a root chain and five child chains corresponding to 5 end effectors (head, left/right arms, left/right legs). HoloPose [7] estimates rotation angles of body joints, and uses it as the prior to guide part-based human mesh reconstruction. Zeng *et al.* [18] designs the continuous UV map to preserve neighboring relationships of the mesh vertices. Zhang *et al.* [19] addresses the occlusion scenario by formulating the task as a UV-map inpainting problem. Since 3DPW is a relatively new benchmark, most literature reported results on Human3.6M, but not 3DPW. We have added their Human3.6M results in Table 1. As we can see, our method outperforms all of the prior works by a large margin.

Recently, researchers are exploring the transformer models for other 3D vision topics, such as multi-view human pose estimation [8] and hand pose estimation based on point could [10]. We encourage the readers to undertake these studies for further explorations.

## B. Implementation Details and Computation Resource

We develop METRO using PyTorch and Huggingface transformer library. We conduct training on a machine equipped with 8 NVIDIA V100 GPUs (32GB RAM) and we use batch size 32. Each epoch takes 32 minutes and we train for 200 epochs. Overall, our training takes 5 days. We use the Adam optimizer and a step-wise learning rate decay. We set the initial learning rate as $1 \times 10^{-4}$ for both

| Model | Dimensionality Reduction Scheme | PA-MPJPE ↓ |
|---|---|---|
| Transformer [16] | $(H+3) \rightarrow 3$ | 208.7 |
| METRO | $(H+3) \rightarrow H/2 \rightarrow 3$ | 192.1 |
| METRO | $(H+3) \rightarrow H/2 \rightarrow H/4 \rightarrow 3$ | 43.8 |
| METRO | $(H+3) \rightarrow H/2 \rightarrow H/4 \rightarrow H/8 \rightarrow 3$ | **36.7** |

Table 1: Performance comparison of different dimentionality reduction schemes, evaluated on Human3.6M validation set. Please note that all the transformer variants have the same total number of hidden layers (12 layers) for fair comparison. $H$=2048.

| | CNN (HRNet-W64) | Transformer |
|---|---|---|
| # Parameters | 128M | **102M** |
| Inference time | 52.05 ms | **28.22 ms** |

Table 2: Number of parameters and inference time per image. The runtime speed is estimated by using batch size 1.

| Positional Encoding | PA-MPJPE ↓ |
|---|---|
| Sinusoidal [16] | 37.5 |
| Ours | **36.7** |

Table 3: Comparison of different positional encoding schemes, evaluated on Human3.6M validation set.

transformer and CNN backbone. The learning rate is decayed by a factor of 10 at the 100th epoch. Our multi-layer transformer encoder is randomly initialized, and the CNN backbone is initialized with ImageNet pre-trained weights. Following [14, 13], we apply standard data augmentation during training.

We evaluate the runtime speed of our model using a machine equipped with a single NVIDIA P100 GPU (16GB RAM). Our runtime speed is about 12 fps using batch size 1. The runtime speed can be accelerated to around 24 fps using batch size 32. Table 2 shows the details of each module in METRO.

For our masked vertex modeling, following BERT [4], we implement it by using a pre-defined special [MASK] token (2051-D floating value vector in our case) to replace the randomly selected input queries.

## C. Progressive Dimensionality Reduction

Since we gradually reduce the hidden sizes in the transformer architecture, one interesting question is whether such a progressive dimensionality reduction scheme is useful. We have conducted an ablation study on different schemes, and Table 1 shows the comparison. In Table 1, the row "(H+3)→3" corresponds to a baseline using one linear projection $H + 3$ to 3. The result is poor. Row "(H+3)→H/2→3" is another baseline which keeps a smaller dimension throughout the network. The result is also bad. Our finding is that large-step (steep) dimension reduction does not work well for 3D mesh regression. Our progressive scheme is inspired by [9] which performed dimensionality reduction gradually with multiple blocks.

## D. Positional Encoding

Since our positional encoding is different from the conventional one, one may wonder what if we use sinusoidal functions [16] but not a template mesh. We have compared our method with the conventional positional encoding which uses sinusoidal functions, and Table 3 shows the results. We see that using sinusoidal functions is slightly worse. This is probably because directly encoding coordinates makes it more efficient to learn 3D coordinate regression.

## E. Qualitative Results

Figure 3 shows a qualitative comparison with the previous image-based state-of-the-art methods [15, 14] in challenging scenarios. These methods only use a single frame as input. In the first row, the subject is heavily bending. Prior works have difficulty in reconstructing a correct body shape for the subject. In contrast, our method reconstructs a reasonable human mesh with correct pose. In the second row, the subject is occluded by the vehicle. We see that prior works are sensitive to the occlusions, and failed to generate correct human mesh. In contrast, our method performs more robustly in this occlusion scenario. In the bottom row, the subject is sitting on the chair. Our method reconstructed a better human mesh compared to the previous state-of-the-art methods.

Figure 4 shows the qualitative results of our method on 3D hand reconstruction. Without making any modifications to the network architecture, our method works well for hands and is robust to occlusions. It demonstrates our method's advantage that it can be easily extended to other types of objects.

| Method | PA-MPVPE ↓ | PA-MPJPE ↓ | F@5 mm ↑ | F@15 mm ↑ |
|---|---|---|---|---|
| I2LMeshNet [15] | 7.6 | 7.4 | 0.681 | 0.973 |
| METRO | 6.7 | 6.8 | 0.717 | 0.981 |
| METRO + Test time augmentation | **6.3** | **6.5** | **0.731** | **0.984** |

Table 4: Effectiveness of test-time augmentation on FreiHAND test set.

## F. Non-local Interactions of Hand Joints

We further conduct quantitative analysis on the non-local interactions among hand joints learned by our model. We randomly sample 1000 samples from FreiHAND test set, and estimate an overall self-attention map. Figure 5 shows the interactions among 21 hand joints. There are 21 rows and 21 columns. Pixel $(i, j)$ represents the amount of attention that hand joint $i$ attends to joint $j$. A darker color indicates stronger attention. We can see that the wrist joint (column 0) receives strong attentions from all the joints. Intuitively wrist joint acts like a "root" of the hand's kinematics tree. In addition, columns 4, 8, 12, and 16 receive strong attentions from many other joints. These columns correspond to the tips of thumb, index, middle, and ring fingers, respectively. These finger tips are end effectors [1] and they can be used to estimate the interior joint positions in inverse kinematics. On the other hand, the tip of pinky only receives attentions from the joints on the ring finger. This is probably because pinky is not as active as the other fingers and its motion is more correlated to the ring finger compared to the other fingers.

## G. Test-Time Augmentation for FreiHAND

We have explored test-time augmentation in our FreiHAND experiments. We do not use test-time augmentation in Human3.6M and 3DPW experiments. Given a test image, we apply different rotations and scaling to the test image. We then feed these transformed images to our model, and average the results to obtain the final output mesh. In order to compute an average 3D mesh, we perform 3D alignment (i.e., Procrustes analysis [6]) to normalize the output meshes. In Table 4, we empirically observed that such an implementation is helpful to improve 0.4 PA-MPVPE on FreiHAND test set.

## H. Limitations

As METRO is a data-driven approach, it may not perform well when the testing sample is very different from the training data. We show some example failure cases in Figure 2 where the test images are downloaded from the Internet. First, as shown in Figure 2(a), we observed that if the target body shape is very different from the existing training data (i.e., SMPL style data), our method may not faithfully reconstruct the muscles of the subject. Secondly, as shown in Figure 2(b), our model fails to reconstruct a correct mesh due to the fact that there is no glove data in the training set. Finally, the proposed method is a mesh-specific approach. If we were to apply our pre-trained right-hand model to the left-hand images, as can be seen in Figure 2(c), our model will not work well. How to develop a unified model for different 3D objects is an interesting future work.

## References


[1] Andreas Aristidou, Joan Lasenby, Yiorgos Chrysanthou, and Ariel Shamir. Inverse kinematics techniques in computer graphics: A survey. In *Computer Graphics Forum*, 2018. 3

[2] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *CVPR*, 2019. 1

[3] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020. 1, 8

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[5] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Kosecka, and Ziyan Wu. Hierarchical kinematic human mesh recovery. *ECCV*, 2020. 1

[6] John C Gower. Generalized procrustes analysis. *Psychometrika*, 1975. 3

[7] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *CVPR*, 2019. 1

[8] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *CVPR*, 2020. 1

[9] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006. 2

[10] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In *ECCV*, 2020. 1

[11] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1

[12] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 1

(a) Example1

Input Output

(b) Example2

Input Output

(c) Example3

Input Output

Figure 2: Failure cases. METRO may not perform well when the testing sample is very different from the training data.


[13] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 1, 2

[14] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 1, 2, 5

[15] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020. 1, 2, 3, 5

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[17] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019. 1

[18] Wang Zeng, Wanli Ouyang, Ping Luo, Wentao Liu, and Xiaogang Wang. 3d human mesh regression with dense correspondence. In *CVPR*, 2020. 1

[19] Tianshu Zhang, Buzhen Huang, and Yangang Wang. Object-occluded human shape and pose estimation from a single color image. In *CVPR*, 2020. 1

Input GraphCMR [14] I2L-M [15] Ours

Figure 3: Qualitative comparison between our method and other single-frame-based approaches. Our method is more robust to challenging poses and occlusions.

Figure 4: Qualitative results of our method on FreiHAND test set. Our method can be easily extended to reconstruct 3D hand mesh.

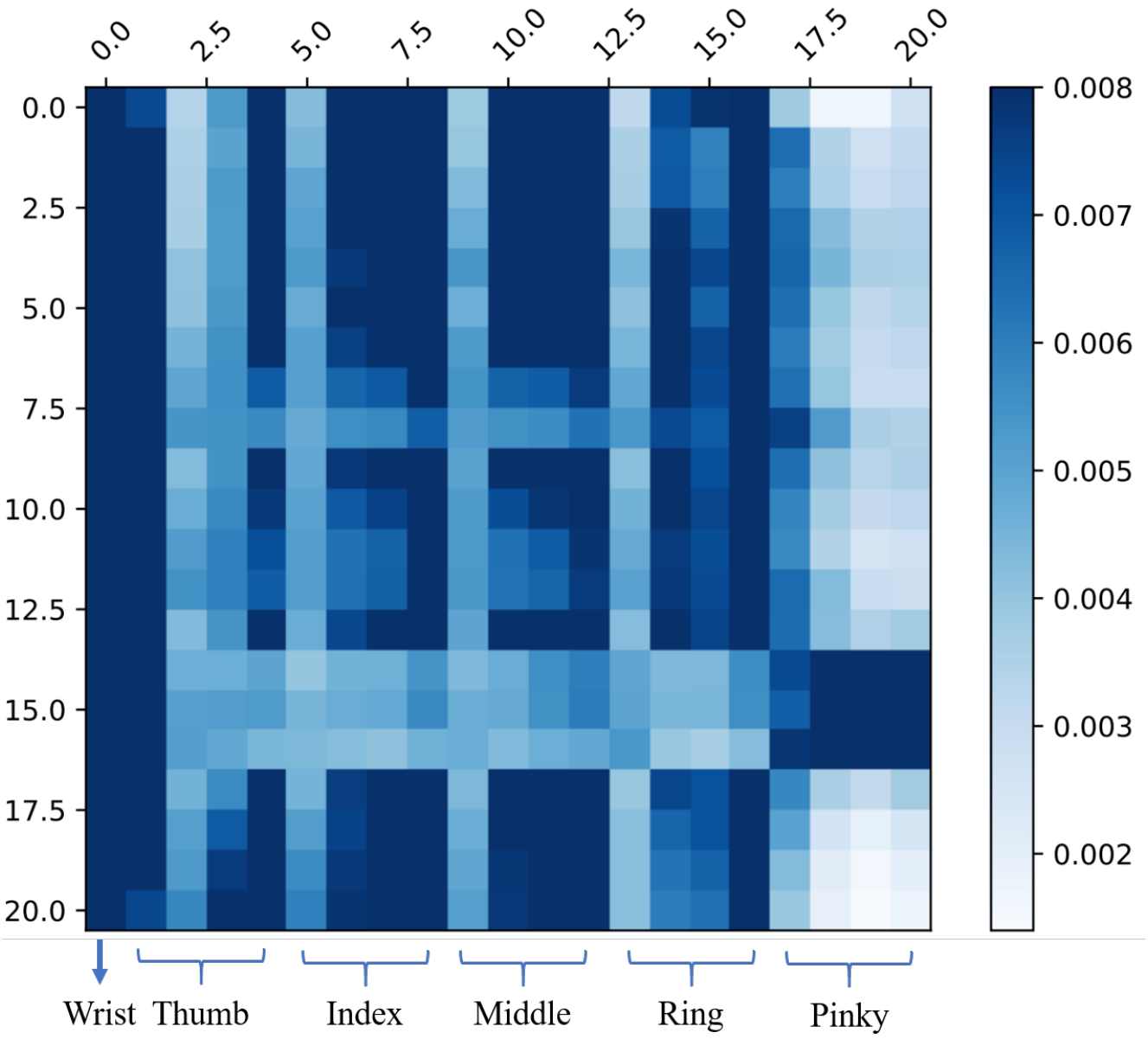


Figure 5: Visualization of self-attentions among hand joints. There are 21 rows and 21 columns corresponding to 21 hand joints. Pixel $(i, j)$ represents the amount of attention that joint $i$ attends to joint $j$. A darker color indicates stronger attention. The definition of the 21 joints is shown in Figure 6.

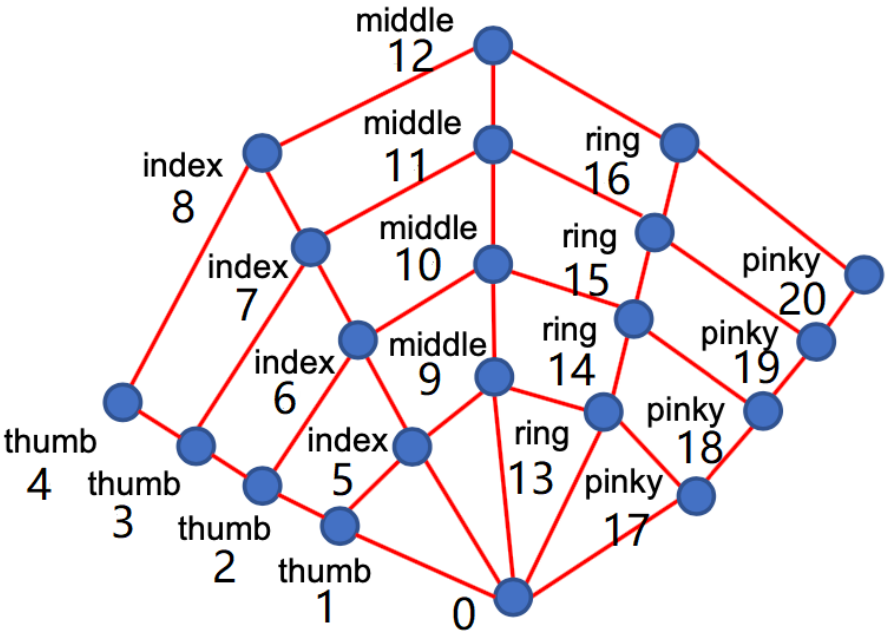


Figure 6: Definition of the hand joints. The illustration is adapted from [3].