HiPART: Hierarchical Pose AutoRegressive Transformer for Occluded 3D Human Pose Estimation

Supplementary Material

A. Pseudo-code for Our HiPART Algorithm

We define the pseudo-code for Stage 1 and Stage 2 of our Hierarchical Pose AutoRegressive Transformer (HiPART) algorithm during training in Alg. 1 and 2.

B. Detailed Coarsening Process

Due to the absence of the hierarchical 2D pose dataset, we construct one as the ground truth for training. We use the pseudo-ground truth 3D mesh provided by Pose2Mesh [7] for Human3.6M. It is widely used in 3D human mesh recovery. Following the dense vertices coarsening method from [2], the process is split into two steps. As shown in Fig. 1, we first progressively coarsen a human mesh graph with 6890 vertices via Heavy Edge Matching (HEM) [4], selecting 96 and 48 joints to represent different levels of human skeleton structure. Then, we project these 3D poses into 2D pixel space and obtain three levels of 2D poses: sparse, dense, and fine, denoted as $\mathbf{x}_s, \mathbf{x}_d, \mathbf{x}_f$, with $\mathbf{x}_s \in \mathbb{R}^{J_s \times 2}$, $\mathbf{x}_d \in \mathbb{R}^{J_d \times 2}$, and $\mathbf{x}_f \in \mathbb{R}^{J_f \times 2}$.

C. Detailed Experimental Setup

We train the MSST with a batch size of 128 for 20 epochs using the AdamW optimizer [13]. The learning rate is initialized at 1e-3 with a weight decay of 0.15, warmed up over the first 500 iterations, and subsequently decayed following a cosine schedule. We set β, λ_{local}, λ_{global}, and τ to 0.25, 1.0, 0.3, and 0.07, respectively. The detailed structure of the encoder is shown in Fig. 2.

We train the HiARM with a batch size of 64 for 50 epochs using the AdamW optimizer. The learning rate is initialized at 5e-4 with a weight decay of 0.03. The λ_d is set to 1.5. The dropout rate of the transformer block is set to 0.25.

For the inference process of HiPART, we select the index with the highest probability from the predicted vectors to generate discrete tokens.

For the lifting stage, we adopt the Adam [9] optimizer. The learning rate is initialized to 1e-3 and decayed by 0.96 per 4 epochs, and we train the model for 25 epochs using a batch size of 256. The overview of the lifting model is shown in Fig. 3.

Our experiments are conducted on one NVIDIA Tesla V100 GPU with the CentOS 7 system, using PyTorch 1.11.0 and Torchvision 0.12.

Algorithm 1 Stage 1: Multi-Scale Skeletal Tokenization (MSST).

Input: The dense and fine 2D poses $\mathbf{x}_d$ and $\mathbf{x}_f$, encoders $\mathcal{E}_d$ and $\mathcal{E}_f$, decoders $\mathcal{D}_s$, $\mathcal{D}_d$ and $\mathcal{D}_f$, sparse and dense codebooks $\mathbf{C}_s$ and $\mathbf{C}_d$, action label $\mathbf{y}_A$, weighting factors β, λ_{local} and λ_{global}, temperature parameter τ, the number of iterations T.

for $t = 0$ to T **do**
  {Forward pass}
  $\mathbf{z}_d \leftarrow \mathcal{E}_f(\mathbf{x}_f)$, $\mathbf{z}_s \leftarrow \mathcal{E}_d(\mathbf{z}_d)$.
  $\mathbf{q}_s \leftarrow \mathcal{Q}(\mathbf{z}_s)$, $\hat{\mathbf{z}}_s \leftarrow \mathbf{C}_s(\mathbf{q}_s)$.
  $\mathbf{z}'_d \leftarrow \mathrm{Concat}(\mathbf{z}_d, \mathcal{D}_s(\hat{\mathbf{z}}_s))$.
  $\mathbf{q}_d \leftarrow \mathcal{Q}(\mathbf{z}'_d)$, $\hat{\mathbf{z}}_d \leftarrow \mathbf{C}_d(\mathbf{q}_d)$.
  $\hat{\mathbf{x}}_d \leftarrow \mathcal{D}_d(\mathbf{q}_d)$, $\hat{\mathbf{x}}_f \leftarrow \mathcal{D}_f(\mathbf{q}_d, \mathcal{D}_s(\mathbf{q}_s))$.
  $\mathbf{p}_A \leftarrow \mathcal{P}_A(\mathrm{Concat}(\hat{\mathbf{z}}_d, \mathcal{D}_s(\hat{\mathbf{z}}_s)))$.
  {Loss calculation}
  Compute the local and global alignment losses according to Eq. 3, 5.
  Compute the Stage 1 loss $\mathcal{L}_1$ based on Eq. 6.
  Update the model based on $\nabla\mathcal{L}_1$.
end for
return

Algorithm 2 Stage 2: Hierarchcal AutoRegressive Modeling (HiARM).

Input: The sparse and dense tokens $\mathbf{q}_s$ and $\mathbf{q}_d$, the sparse 2D pose $\mathbf{x}_s$, the weighting factor λ_d, the number of iterations T.

for $t = 0$ to T **do**
  {Forward pass}
  $\hat{\mathbf{z}}_s \leftarrow \mathbf{C}_s(\mathbf{q}_s)$, $\hat{\mathbf{z}}_d \leftarrow \mathbf{C}_d(\mathbf{q}_d)$
  $\{\mathbf{g}_j^i\}_{j=0,1,\ldots,r} \leftarrow \mathrm{LSAB}(\hat{\mathbf{z}}_s^i, \hat{\mathbf{z}}_d^{(i,1)}, \ldots, \hat{\mathbf{z}}_d^{(i,r)})$
  $\mathbf{g}^i = \mathrm{avg}(\mathbf{g}_0^i, \mathbf{g}_1^i, \ldots, \mathbf{g}_r^i)$
  $\{\mathbf{h}^k\}_{k=0,1,\ldots,i} \leftarrow \mathrm{GCSAB}(\mathbf{g}^0, \mathbf{g}^1, \ldots, \mathbf{g}^i)$
  $\{p_j^{i+1}\}_{j=0,1,\ldots,r} \leftarrow \mathrm{PH}(\mathbf{h}^i, \hat{\mathbf{z}}_s^i, \ldots, \hat{\mathbf{z}}_s^i)$
  {Loss calculation}
  Compute the Stage 2 loss $\mathcal{L}_2$ based on Eq. 12.
  Update the model based on $\nabla\mathcal{L}_2$.
end for
return

D. Additional Discussion on Related Work

Hierarchical AutoregRessive Models. VQ-VAE [19] has pioneered a two-stage image generation process, which involves initially quantizing images into discrete tokens,

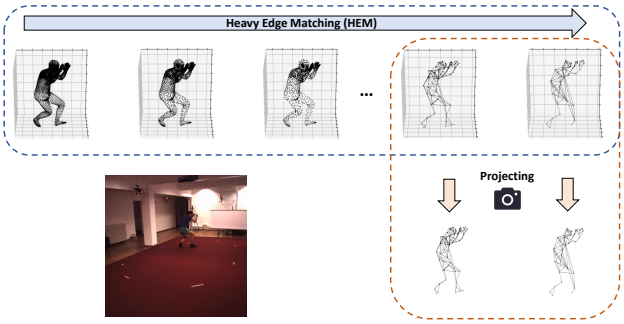


Figure 1. The overview of the coarsening process, consisting of two steps. We first progressively coarsen a human mesh with 6890 vertices via Heavy Edge Matching (HEM) [4]. Then we project these 3D poses into 2D pixel space.

followed by their reconstruction in the subsequent stage. Based on VQ-VAE, many subsequent works leverage hierarchical discrete tokens for coarse-to-fine image generation. For instance, VQ-VAE-2 [16] uses models of different sizes for top and bottom tokens, while Hierarchical VQ-VAE [15] creates two levels of tokens to disentangle structural and textural image information. Our method differs from these works in three key aspects: **(1)** Human skeletons, with their non-Euclidean structure, require a tailored model and regression prediction order distinct from those used in conventional image data. **(2)** Compared to the unengaged hierarchical tokens in image generation, we give specific meanings (*i.e.*, representing the multi-level 2D poses) to the multi-scale discrete tokens with the corresponding constraint. **(3)** While hierarchical tokens in image generation balance code sequence length with image quality, our tokens provide multi-scale skeletal context specifically designed to tackle occlusions.

Discrete Representation Models in 3D HPE. Recently, several 3D human pose estimation (HPE) methods have adopted the two-stage approach to learn discrete representations. PCT [6] establishes a framework that learns a discrete codebook and then treats pose estimation as a classification problem. However, this classification approach fails to efficiently capture the latent distribution of discrete tokens. Di^{2}Pose [20] employs a diffusion model to generate discrete tokens, enhancing prediction accuracy, but suffers from slow inference speed due to the need for numerous sampling steps. In contrast, our method introduces a hierarchical autoregressive modeling scheme for faster and more reliable predictions. Moreover, instead of directly generating 3D poses, our approach focuses on producing hierarchical dense 2D poses in a two-stage process.

Diffusion Models in 3D HPE. In recent years, diffusion models have also been increasingly applied in 3D HPE. For instance, diffusion models are employed to progressively refine the pose distribution, reducing uncertainty from high

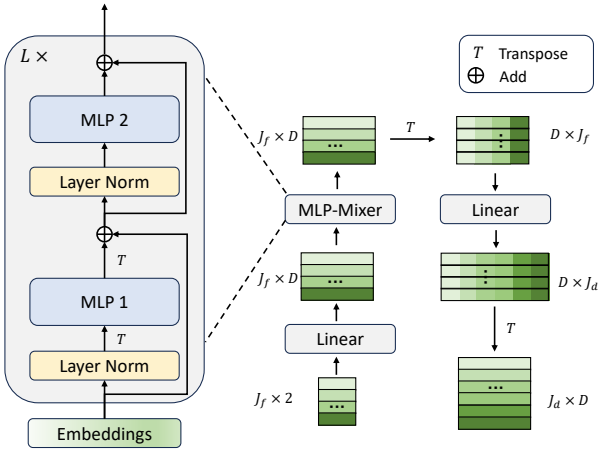


Figure 2. The detailed structure of the encoder of MSST, taking $\mathcal{E}_f$ as the example. Following PCT [6], the fine pose is first fed to a linear projection layer to transform the embedding dimension. Subsequently, these enhanced embeddings are passed through L MLP-Mixer blocks [18], which deeply fuse the pose feature. We can finally obtain encoded embeddings by applying a linear projection along the joint axis and transposing the embeddings.

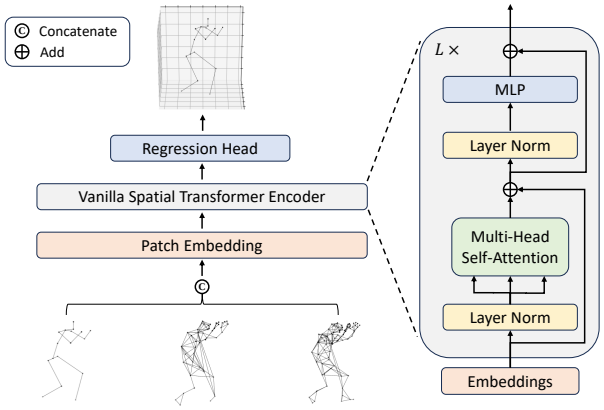


Figure 3. The overview of the lifting model, consisting of patch embedding, vanilla spatial transformer encoder, and regression head. We concatenate three levels of 2D poses along the joint axis as the input to patch embedding and vanilla spatial transformer encoder. Then the corresponding embeddings are regressed into the target 3D pose.

to low throughout the estimation process [3, 5, 7]. Other approaches leverage diffusion models to generate multiple pose hypotheses from a single 2D observation [8, 17], effectively addressing ambiguity in pose estimation. However, these models typically exhibit lower throughput and slower inference speeds compared to the autoregressive approach in our method, due to the extensive sampling steps needed for precision. We further explore this in our experiments detailed in Section E.

Table 1. Results on the H36M test set under occlusion (*i.e.*, mask and crop).

Mask Ratio	0.0	0.2	0.4	0.6	0.8
DiffPose [7]	49.7	64.2△14.5	83.9△34.2	140.3△90.6	284.6△234.9
Ours	42.0	53.2△11.2	77.5△35.5	125.1△83.1	269.4△227.4
Crop Ratio	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**
DiffPose [7]	49.7	50.0△0.3	50.9△1.2	58.5△8.8	72.7△23.0
Ours	42.0	42.2△0.2	42.8△0.8	48.4△6.4	61.5△19.5

Table 2. Comparison with diffusion models in 3D HPE. We compare inference speed (frame per second (FPS)), and MPJPE on Human3.6M.

Method	FPS ↑	MPJPE ↓
DiffPose [7]	173	49.7
DiffuPose [3]	188	49.4
vanilla spatial transformer w./ ours	396	42.0
MixSTE [22] ($f = 81$) w./ ours	**681**	**39.3**

E. Additional Experiment Results

In this section, we conduct a series of additional experiments on Human3.6M to further demonstrate the effectiveness of our method.

Results on Human3.6M under Occlusion. To validate the performance of our method under different occlusion conditions, we synthesize the occlusion scenarios by masking or cropping the test images. We investigate the cascaded pyramid network (CPN) [1] with a ResNet-50 [21] backbone as the 2D keypoint detector to infer the 2D poses of the test set. We load the model weight from [14], which is pretrained on COCO [12]. After obtaining the 2D results, we compare the performance of our method with DiffPose [7]. Tab. 1 shows that our method performs better under different occlusion conditions and exhibits stronger robustness when the occlusion worsens.

Comparison with Diffusion Models in 3D HPE. To compare our autoregressive method with diffusion-based methods including DiffPose [7] and DiffuPose [3], we compare the inference speed and MPJPE in Tab. 2. Results show that our method incorporating a vanilla spatial transformer significantly outperforms these two diffusion models on FPS and MPJPE, demonstrating the accuracy and efficiency of our autoregressive method. Further integrating into the temporal-based method, *i.e.*, MixSTE [22], achieves the best inference speed and prediction accuracy.

Detailed Densification Results. To evaluate the effectiveness of the hierarchical dense 2D poses, we conduct a toy experiment on Human3.6M with the ground truth 2D sparse pose. As shown in the top of Fig. 4, adding the ground truth hierarchical 2D dense poses into a vanilla spatial transformer brings a 20.1 mm improvement of MPJPE. In real-world applications, our method achieves the best

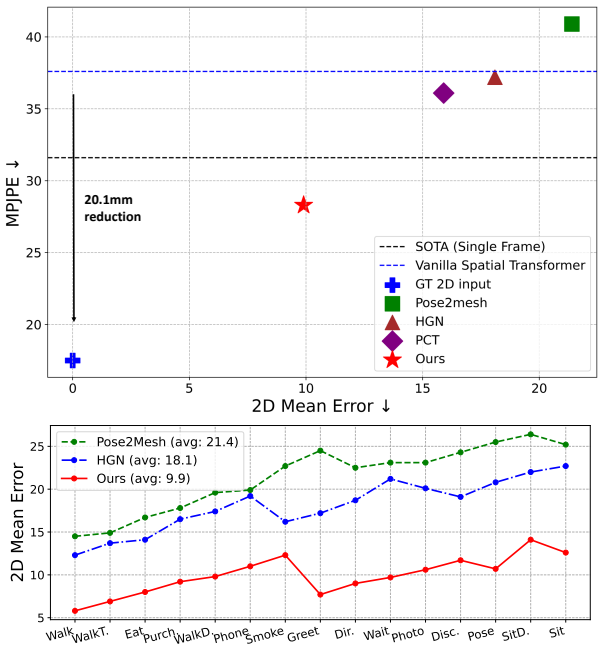


Figure 4. **Top:** The prediction accuracy of the fine 2D pose (96 joints) and lifted 3D pose (17 joints) across various methods. **Bottom:** Detailed densification results across various actions for three methods on Human3.6M using the ground truth sparse 2D pose (*i.e.*, 17-joint pose) as the input.

performance in 2D mean error and MPJPE compared with Pose2Mesh [2], HGN [10], and PCT [6]. Furthermore, The bottom of Fig. 4 illustrates the 2D Mean Error of the fine 2D pose across various actions for three different methods. Two conclusions can be intuitively concluded: (1) As the complexity of the action increases, *i.e.*, when there is a higher frequency of occlusions, the densification performance deteriorates. (2) Our method enhances the densification performance, with an average improvement of 11.5mm for Pose2Mesh and 8.2mm for HGN. This enhancement is particularly evident for actions with severe occlusions, such as the Sit action, where our method achieves an improvement of 13.2mm for Pose2Mesh and 11.2mm for HGN.

Ablation Study on Different Parameters of HiPART. Tab. 3 details the effects of various parameters on our model's performance and complexity. Optimal results are achieved with 4 MLP-Mixer blocks of the MSST encoder and 12 blocks in GCSAB, with no significant improvements from adding more layers. Additionally, the results show that increasing the embedding dimension from 96 to 128 enhances performance, but dimensions larger than 128 do not yield further benefits. Therefore, we establish the default settings as $L_1 = 4$, $L_2 = 12$, and $D_1 = D_2 = 128$.

Extra Sequence-based Results. Table 4 provides extra results for sequence-based DiffPose and MixSTE with $f = 243$. Our method is best in MPJPE under both frame and

Table 3. Ablation study on different parameters of HiPART. L_1 and L_2 denote the number of blocks of the MSST encoder and GCSAB, respectively. D_1 and D_2 are the embedding dimensions of the MSST encoder and HiARM, respectively.

L_1	L_2	D_1	D_2	Params(M)	FLOPs(G)	MPJPE
2	6	128	128	1.8	1.41	44.1
3	8	128	128	2.1	1.82	43.5
4	12	128	128	2.4	2.24	**42.0**
5	16	128	128	2.7	2.65	42.2
4	12	96	96	2.0	1.77	43.9
4	12	128	128	2.4	2.24	**42.0**
4	12	256	256	3.6	4.14	42.5

Table 4. Comparison with DiffPose and MixSTE on Human3.6M under frame (**left**) and sequence (**right**) based settings.

Method	MPJPE	FPS	Method	MPJPE	FPS
MixSTE (f=1)	51.1	358	MixSTE (f=243)	40.9	**1055**
DiffPose (f=1)	49.7	173	DiffPose (f=243)	36.9	671
Ours (f=1)	**42.0**	**396**	Ours+MixSTE (f=243)	**36.7**	577

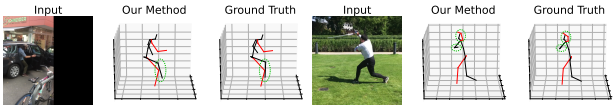


Figure 5. Failure cases on 3DPW. **Left:** severe occlusion where even human perception struggles to infer 3D poses. **Right:** unseen fencing motion which is absent from the training set.

sequence based settings.

F. Additional Visualization Analysis

In this section, we provide additional visualization analysis to better understand our approach.

Discussion of Failure Cases. Fig. 5 shows the failure cases on 3DPW. (*i.e.*, severe occlusion and rare poses).

More Analysis on the Similarity of Hierarchical Codebooks. We supplement the cosine similarity matrices calculated from the randomly selected 100 tokens. As shown in Fig. 6, we can draw similar conclusions to those in the main text.

More Visualization of Hierarchical Poses. Qualitative results in Fig.7 further demonstrate that additional joints around occluded areas provide richer skeletal information. For instance, when the right leg is occluded as shown in the fourth row of Fig. 7, the sparse pose offers limited support with only two joints (knee and ankle) available, while the denser pose includes multiple leg joints, capturing a more detailed structure around the occlusion and aiding in predicting the occluded right leg.

More Visualization of Poses under Occlusion. As shown in Fig.8, we provide additional qualitative results on Human3.6M and 3DPW, comparing our method with DiffPose[7]. DiffPose can predict decent results on Human3.6M, but its predictions in occluded areas become significantly poorer when generalized to the more occlusion-heavy 3DPW dataset. In contrast, our method maintains strong occlusion robustness on 3DPW. For instance, in the first row of the 3DPW data, where the back severely occludes both arms, DiffPose's predictions for the occluded area deviate substantially from the ground truth. Our approach leverages hierarchical information near the arms to aid in inference, effectively predicting the joints at the occluded locations.

G. Limitation and Future Work

A current limitation of our approach is its reliance on the single-frame based methods [11] for the lifting model. Applying our method directly to the temporal-based lifting models [22] would slow down inference, limiting our ability to further utilize temporal information and indicating room for improvement in our approach. This is attributed to the expansion of input joint quantities. The conventional attention mechanisms have a computational complexity that scales significantly with the joint quantities.

Moving forward, we aim to develop a temporal-based lifting model compatible with hierarchical 2D poses. Our future work will involve strategies such as sampling the key joints from hierarchical 2D poses and optimizing the computation of attention mechanisms to manage the joint quantity increase effectively. By doing so, we expect to integrate the strengths of temporal-based models with our hierarchical pose approach, thereby improving the accuracy of pose estimation while maintaining computational efficiency.

Besides, Tab. 6 in the main paper shows that our method greatly improves single-frame lifting models more than multi-frame methods, suggesting considerable potential for optimizing how our densification approach combines with temporal information, which is straightforward in our experiments. Firstly, each frame of the 2D pose sequence is fed into HiPART to generate hierarchical 2D poses. These poses are concatenated along the temporal and joint dimensions to form a tensor of size $T \times (J_s + J_d + J_f) \times 2$, which is input for temporal-based lifting models to infer the final 3D pose. In the future, we plan to delve into more effective densification methods for integrating hierarchical spatial and temporal information to better exploit their interplay and further boost the performance of 3D HPE.

References

[1] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018. 3

[2] 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 *Proceed-*

ings of the 16th European Conference on Computer Vision, pages 769–787, 2020. 1, 3

[3] Jeongjun Choi, Dongseok Shim, and H Jin Kim. DiffuPose: Monocular 3D human pose estimation via denoising diffusion probabilistic model. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3773–3780, 2023. 2, 3

[4] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29*, pages 3837–3845, 2016. 1, 2

[5] Runyang Feng, Yixing Gao, Tze Ho Elden Tse, Xueqing Ma, and Hyung Jin Chang. DiffPose: Spatiotemporal diffusion model for video-based human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14861–14872, 2023. 2

[6] Zigang Geng, Chunyu Wang, Yixuan Wei, Ze Liu, Houqiang Li, and Han Hu. Human pose as compositional tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 660–671, 2023. 2, 3

[7] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. DiffPose: Toward more reliable 3D pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 2, 3, 4, 8

[8] Karl Holmquist and Bastian Wandt. Diffpose: Multi-hypothesis human pose estimation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15977–15987, 2023. 2

[9] Diederik P Kingma. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations*, 2015. 1

[10] Han Li, Bowen Shi, Wenrui Dai, Yabo Chen, Botao Wang, Yu Sun, Min Guo, Chenlin Li, Junni Zou, and Hongkai Xiong. Hierarchical graph networks for 3D human pose estimation. In *British Machine Vision Conference*, 2021. 3

[11] Han Li, Bowen Shi, Wenrui Dai, Hongwei Zheng, Botao Wang, Yu Sun, Min Guo, Chenglin Li, Junni Zou, and Hongkai Xiong. Pose-oriented transformer with uncertainty-guided refinement for 2D-to-3D human pose estimation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 1296–1304, 2023. 4

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision*, pages 740–755, 2014. 3

[13] I Loshchilov. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations*, 2019. 1

[14] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 3

[15] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021. 2

[16] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems 32*, pages 14866–14876, 2019. 2

[17] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3D human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14761–14771, 2023. 2

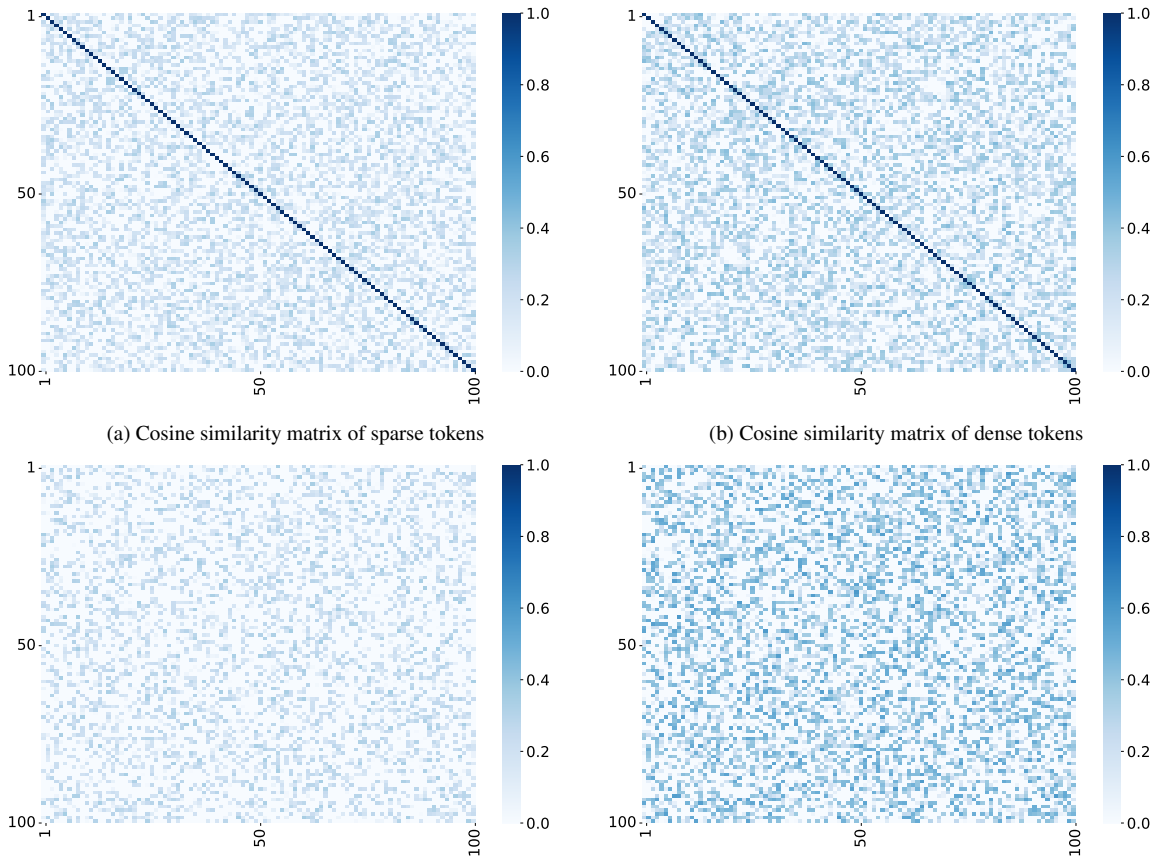
[18] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP architecture for vision. In *Advances in Neural Information Processing Systems 34*, pages 24261–24272, 2021. 2

[19] Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30*, pages 6309–6318, 2017. 1

[20] Weiquan Wang, Jun Xiao, Chunping Wang, Wei Liu, Zhao Wang, and Long Chen. Di2Pose: Discrete diffusion model for occluded 3D human pose estimation. In *Advances in Neural Information Processing Systems 37*, pages 98717–98741, 2024. 2

[21] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 3

[22] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. MixSTE: Seq2seq mixed spatio-temporal encoder for 3D human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13232–13242, 2022. 3, 4



(a) Cosine similarity matrix of sparse tokens

(b) Cosine similarity matrix of dense tokens

(c) Cosine similarity matrix of sparse and dense tokens (w./o. Local Alignment)

(d) Cosine similarity matrix of sparse and dense tokens (w./ Local Alignment)

Figure 6. Hierarchical codebooks similarity analysis. The cosine similarity is calculated based on the **random selection of 100 tokens** from both the sparse and dense codebooks. (c) (d) The x-axis represents sparse tokens, and the y-axis represents dense tokens.

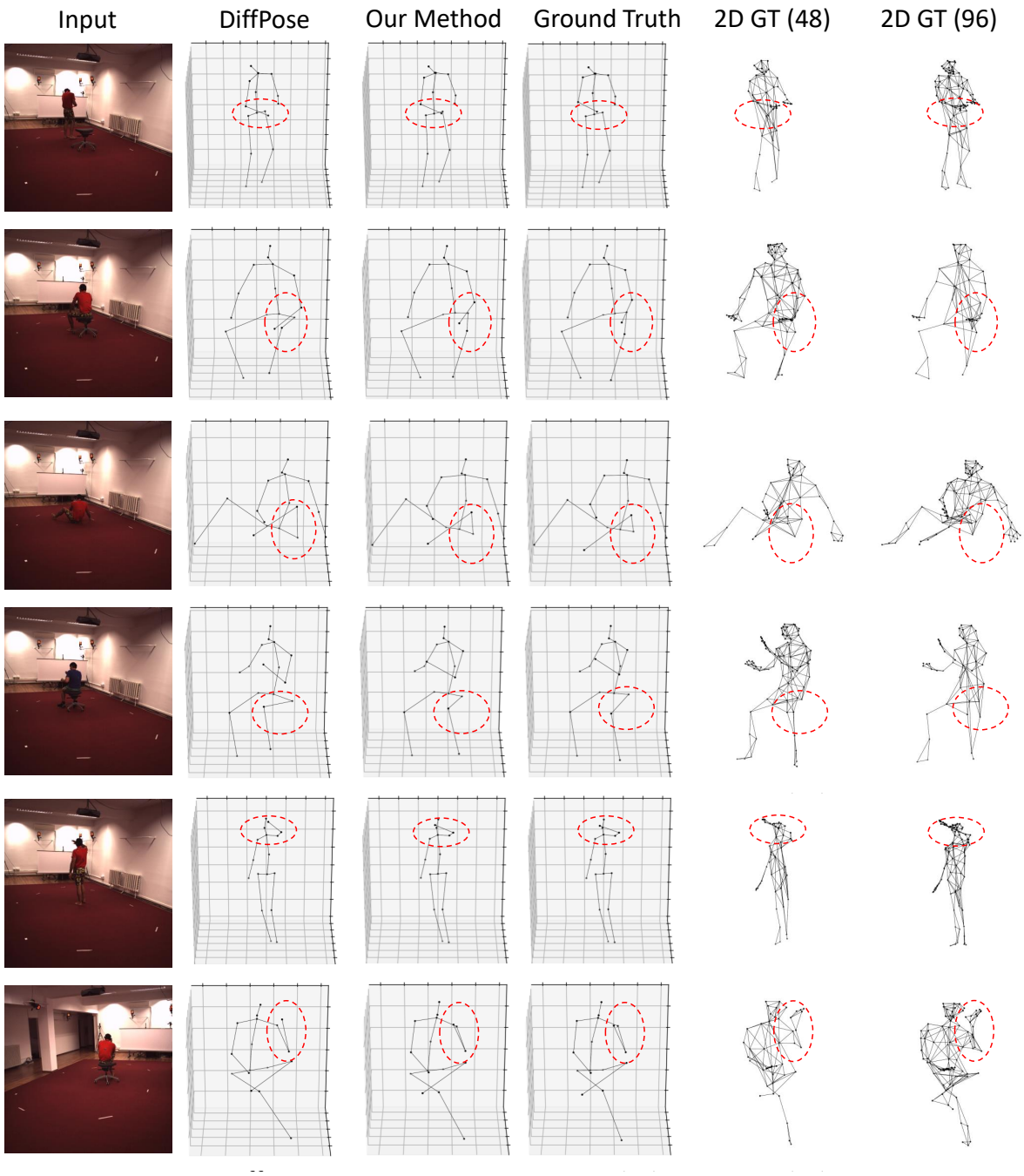


Figure 7. Qualitative results of reconstructed 3D poses and hierarchical 2D poses on Human3.6M under occlusions.

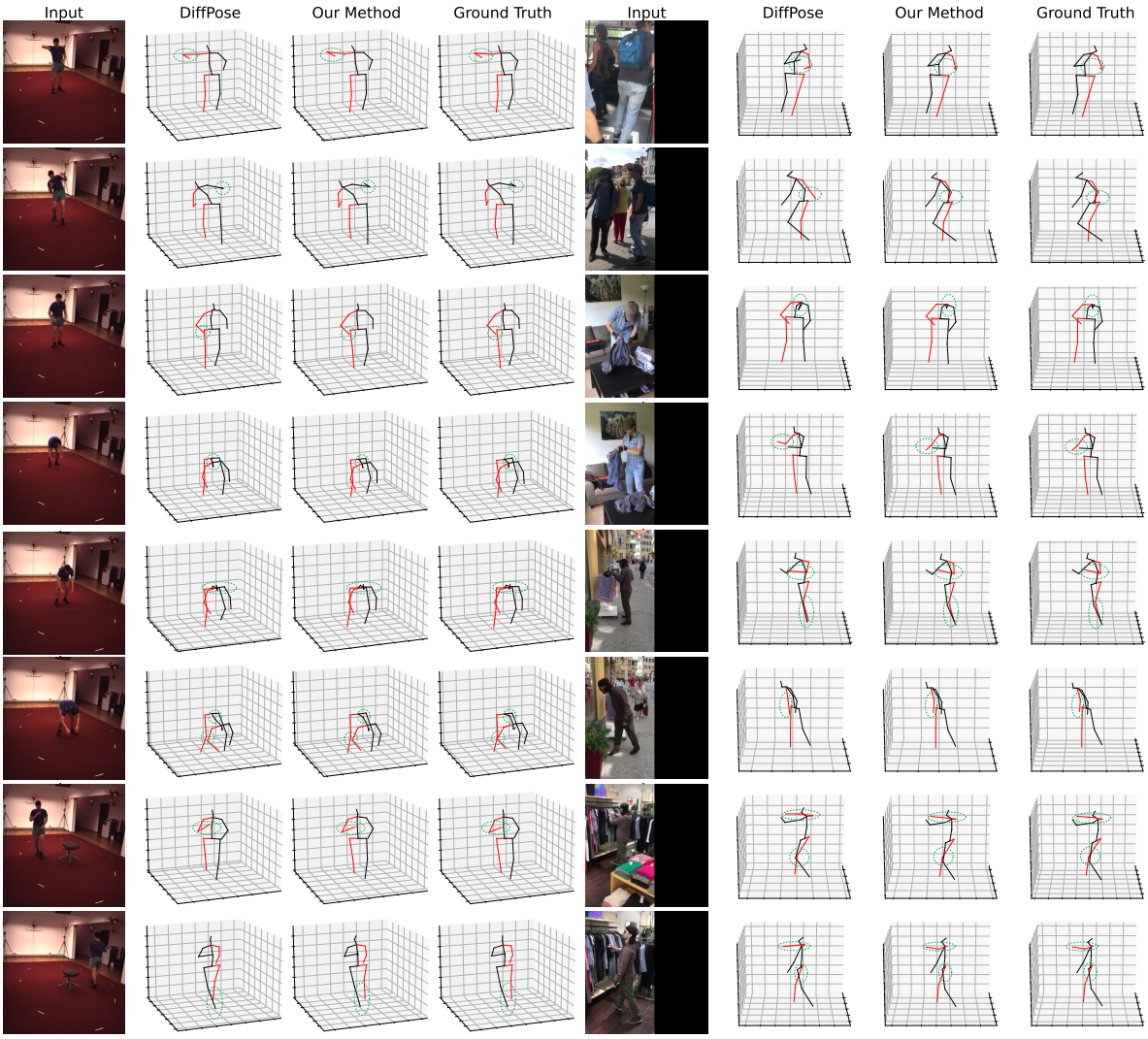


Figure 8. Qualitative results compared with DiffPose [7] on Human3.6M (**left**) and 3DPW (**right**).