Point2Pose: A Generative Framework for 3D Human Pose Estimation with Multi-View Point Cloud Dataset

Supplementary Document

Hyunsoo Lee[1*] Daeum Jeon[2,3] Hyeokjae Oh[2,3]
[1] ECE, Seoul National University [2] CS, KAIST [3] Soulart Inc.

philip21@snu.ac.kr, {daumi,boerck}@kaist.ac.kr

Appendix

A. Pseudo-code of Point2Pose

We explain the general training procedure of Point2Pose$_{\mathrm{DM}}$ in Algorithm 1, and inference process in Algorithm 2.

Algorithm 1 Training Point2Pose with DM

Inputs: Sequential point clouds $\mathbf{P}$, Ground-truth pose data $\mathbf{X}^{\mathrm{gt}}$, sampling steps I
repeat
  Compute $\mathbf{F}_{px}$ using Eq. (5)
  Sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $i \sim \mathrm{Uniform}([0, I])$
  Calculate $\tilde{\mathbf{X}}_i \leftarrow \mathrm{Perturb}_{\mathrm{DM}}(\mathbf{X}^{\mathrm{gt}}, i)$
  Calculate $\mathbf{X}_i^{\mathrm{pred}}$ using Eq. (7)
  Calculate $\mathcal{L}$ using Eq. (14)
  Optimize θ with SGD on $\nabla_\theta \mathcal{L}$
until converged
Output: Optimized Point2Pose parameter θ^*

Algorithm 2 Sampling $\mathbf{X}^{\mathrm{pred}}$ from Point2Pose with DM

Inputs: Sequential point clouds $\mathbf{P}$, sampling steps I, optimized Point2Pose parameters θ^*
Compute $\mathbf{F}_{px}$ using Eq. (5)
Sample $\tilde{\mathbf{J}}_I \sim \mathcal{N}(0, \mathbf{I})$, $\tilde{\mathbf{R}}_I \sim \mathcal{N}(0, \mathbf{I})$
for $i \leftarrow I, \cdots, 1$ **do**
  Calculate $\tilde{\mathbf{J}}_{i-1}$ using Eq. (15)
end for
for $i \leftarrow I, \cdots, 1$ **do**
  Calculate $\tilde{\mathbf{R}}_{i-1}$ using Eq. (16)
end for
$\mathbf{X}^{\mathrm{pred}} \leftarrow [\mathbf{J}_0^{\mathrm{pred}}, \mathbf{R}_0^{\mathrm{pred}}]$
Output: Sampled pose data $\mathbf{X}^{\mathrm{pred}}$

* indicates the corresponding author.

B. Additional ablation study

To show the effectiveness of the spatio-temporal point cloud encoder, we conduct an additional ablation study. Especially, we set $T = 1$ for the point cloud encoder and train Point2Pose$_{\mathrm{OT-CFM}}$ model using the ITOP-Side [2] dataset. Notably, as shown in Table 11, the performance drops when only the final frame of the point cloud is used, demonstrating the necessity of a spatio-temporal point cloud encoder module.

Table 11. Additional ablation study on the temporal encoding of the point cloud encoder using the ITOP-Side [2] dataset.

Method	ITOP-Side [2]
	MPJPE (↓)
Point2Pose$_{\mathrm{OT-CFM}}$ ($T = 4 \rightarrow T = 3$)	**125.01**
Point2Pose$_{\mathrm{OT-CFM}}$ (only $T = 1$ for PC encoder, $T = 4 \rightarrow T = 3$)	154.35

C. Additional qualitative results

We visualize additional results of our method and MOVIN [3] on the LiDARHuman26M [6] and LIPD [6] dataset in Figure 6 and 7, respectively. As shown, our method achieves outstanding performance compared to MOVIN.

D. Graph-based point cloud clustering

In this section, we discuss about the *graph-based point cloud clustering (GPC)* mentioned in Section 5.1, which is a novel strategy used for eliminating noise in the LiDARHuman26M [4] dataset.

Point clouds which captures humans in a real world often contains noisy points, due to the technical limitations of capture devices. To address the issue, we propose a novel point cloud pre-processing method which effectively eliminates noisy points using graph-based clustering. We firstly generate 2D KD-tree $\mathcal{D}$ by projecting input 3D point clouds into xy-plane, and extract n_C 2D point clusters $\{\mathcal{C}_i\}_{i=0}^{n_C}$

with BFS. Then, to eliminate noisy points, we modify each clusters by

$$\tilde{\mathcal{C}}_i = \{x_j | x_j \in \mathcal{C}_i, |z(x_j) - \mu(\mathcal{C}_i)| < \lambda \cdot \sigma(\mathcal{C}_i)\},$$
$$\mathcal{C}_i = \tilde{\mathcal{C}}_i \text{ if } |\tilde{\mathcal{C}}_i| \geq n_c \text{ else } \phi \tag{19}$$

where $\mu(\mathcal{C}_i)$ and $\sigma(\mathcal{C}_i)$ denotes the mean and standard deviation of depth of points consisting $\mathcal{C}_i$, $z(\cdot)$ denotes depth of given point, and λ and n_c is hyperparameter. Next we select the largest cluster $\mathcal{C}_{i*}$, where $i^* = argmax_i |\mathcal{C}_i|$. Based on $\mathcal{C}_{i*}$, *reachable clusters* $\mathcal{C}_{\text{reach}}$ are calculated by

$$\mathcal{C}_{\text{reach}} = \{\mathcal{C}_i |^{\exists} x_j \in \mathcal{C}_{i*}, x_k \in \mathcal{C}_i \text{ s.t. } x_k \in \text{kNN}(x_j), \\ M_1 < |z(x_k)| < M_2, \;\; i \neq i^*\}, \tag{20}$$

where M_1 and M_2 are threshold derived by the depth values of points of $\mathcal{C}_{i^*}$, and $\text{kNN}(\cdot)$ returns k-nearest neighbors of given point. We finally construct clustered points as follows:

$$\mathcal{C}_{\text{output}} = \{x_i | x_i \in \{\mathcal{C}_{i^*}\} \cup \mathcal{C}_{\text{reach}}\}. \tag{21}$$

Effectiveness of GPC. Using LiDARHuman26M dataset, we measured chamfer distance (CD) between human mesh $\mathbf{m}$ reconstructed with SMPL [5] and point cloud $\mathbf{p}$, which is calculated by

$$\frac{1}{|\mathbf{p}|} \sum_{x_1 \in \mathbf{p}} \min_{x_2 \in \mathbf{m}} \|x_1 - x_2\|_2^2 + \frac{1}{|\mathbf{m}|} \sum_{x_1 \in \mathbf{m}} \min_{x_2 \in \mathbf{p}} \|x_1 - x_2\|_2^2. \tag{22}$$

Notably, chamfer distance is decreased from 36.55 to 34.70 by utilizing Graph-based point cloud clustering, which demonstrates its effectiveness on noisy points elimination.

E. Experimental details

Since all of the LiDARHuman26M [4], LIPD [6] and MVPose3D datasets do not provide joint velocity and acceleration data, we use joint coordinates and rotations to train both Point2Pose and MOVIN. To train Point2Pose on the LiDARHuman26M and LIPD dataset, we used 30 epochs, halving the learning rate after 20 epochs. In case of training the proposed method on MVPose3D dataset, we use 20 epochs. MOVIN is trained for 120 epochs on LiDARHuman26M and LIPD, and for 60 epochs on MVPose3D dataset, with a learning rate of 10^{-4}. We do not utilize learning rate scheduling when training MOVIN. Note that we reproduce MOVIN [3] due to the inaccessibility of its official implementation. For validation, we include our implementation of MOVIN in the supplementary material. We use single-view point clouds from the MVPose3D for all experiments.

F. Details on the proposed dataset: MVPose3D

F.1. Dataset visualization

We visualize samples from the MVPose3D dataset in Figure 8, 9. Each sample consists of four point clouds, five RGB images captured from different viewpoints, and motion data including joint coordinates and rotations. As shown in Figure, our dataset involves diverse human motions and dense point clouds. Note that we mask each participant's identity only for visualization.

F.2. Data preprocessing

To segment point cloud corresponding to the human body, we first eliminate the background using a reference depth map, then apply DFS [8] for initial segmentation. Next, we convert the segmented depth map into a point cloud, and multiply the transform matrix obtained through extrinsic calibration. We further refine the segmented point clouds using DBSCAN [1] and Statistical Outlier Removal [7]. The overall process is summarized in Algorithm 3.

Algorithm 3 Depth-based human body segmentation

Input: Captured depth map $\mathbf{D} \in \mathbb{R}^{M \times N}$, reference map $\mathbf{D}_{\text{ref}} \in \mathbb{R}^{M \times N}$, iToF extrinsic parameters $\mathbf{T}_{\text{ext}}$, hyperparameter τ
for $u \leftarrow 1, \cdots, M$ **do**
  for $v \leftarrow 1, \cdots, N$ **do**
    if $\left|\mathbf{D}(u,v) - \mathbf{D}_{\text{ref}}(u,v)\right| < \tau$ **then**
      $\mathbf{D}(u,v) \leftarrow 0$
    end if
  end for
end for ▷ Background elimination
$\mathbf{C} \leftarrow \text{Largest\_connected\_component}(\mathbf{D})$
$\mathbf{P} \leftarrow \text{Depth\_to\_point\_cloud}(\mathbf{C})$
$\mathbf{P}_{\text{transformed}} = \left\{ \mathbf{T}_{\text{ext}} \left(x, y, z, 1\right)^\top \;\middle|\; (x, y, z) \in \mathbf{P} \right\}$
$\mathbf{P}_{\text{DB}} \leftarrow \text{DBSCAN}\left(\mathbf{P}_{\text{transformed}}\right)$
$\widetilde{\mathbf{P}} \leftarrow \text{SOR}\left(\mathbf{P}_{\text{DB}}\right)$ ▷ Human body segmentation
Output: Clustered human body point cloud $\widetilde{\mathbf{P}}$

F.3. Details on human resources

Motion capture from subjects. MVPose3D contains 18 distinct actions performed by each subject, covering a wide range of human motions. Table 12 summarizes the details of captured motions.

Table 12. Types of human actions included in MVPose3D dataset.

Static	Locomotive
T-pose, A-pose, Attention	leg, side gymnastics
walking in place, throwing	back, jumping gymnastics
Upper-body T-pose and rotate	double arm-leg gymnastics
kick, boxing	running in a straight line
sit-to-stand	running in a circle
Jump while rotating 180°	Boxing while running

Potential risks related to study participants. There are no potential risks to participants from the data acquisition experiment itself, since it does not involve any equipment or elements that are harmful to humans. This information has been informed to the participants in advance.

G. Computational overhead

Using a single NVIDIA GeForce RTX 3090 GPU, our method takes 1.62 hours and requires 19.46 GB of GPU memory to train for 30 epochs with a batch size of 32 on the LIPD [6] dataset.

H. Social impacts

This work can be widely applied across various fields that require precise 3D human pose estimation, such as virtual reality (VR), augmented reality (AR), digital healthcare, and online gaming. However, due to the limitations of the generative model (DM and OT-CFM) used, it may produce inaccurate or harmful results, which may potentially pose a risk to users.

I. Limitations

Due to the limitations of generative models, our method can produce implausible results for extremely abnormal poses. In addition, the pose estimation error may increase when dealing with very sparse point cloud data.

References

[1] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 2

[2] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *ECCV*, 2016. 1

[3] Deok-Kyeong Jang, Dongseok Yang, Deok-Yun Jang, Byeoli Choi, Taeil Jin, and Sung-Hee Lee. Movin: Real-time motion capture using a single lidar. In *Computer Graphics Forum*, 2023. 1, 2, 4, 5

[4] Jialian Li, Jingyi Zhang, Zhiyong Wang, Siqi Shen, Chenglu Wen, Yuexin Ma, Lan Xu, Jingyi Yu, and Cheng Wang. Lidar-cap: Long-range marker-less 3d human motion capture with lidar point clouds. In *CVPR*, 2022. 1, 2, 4

[5] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, 2015. 2

[6] Yiming Ren, Chengfeng Zhao, Yannan He, Peishan Cong, Han Liang, Jingyi Yu, Lan Xu, and Yuexin Ma. Lidar-aid inertial poser: Large-scale human motion capture by sparse inertial and lidar sensors. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 1, 2, 3, 5

[7] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 2010. 2

[8] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1972. 2

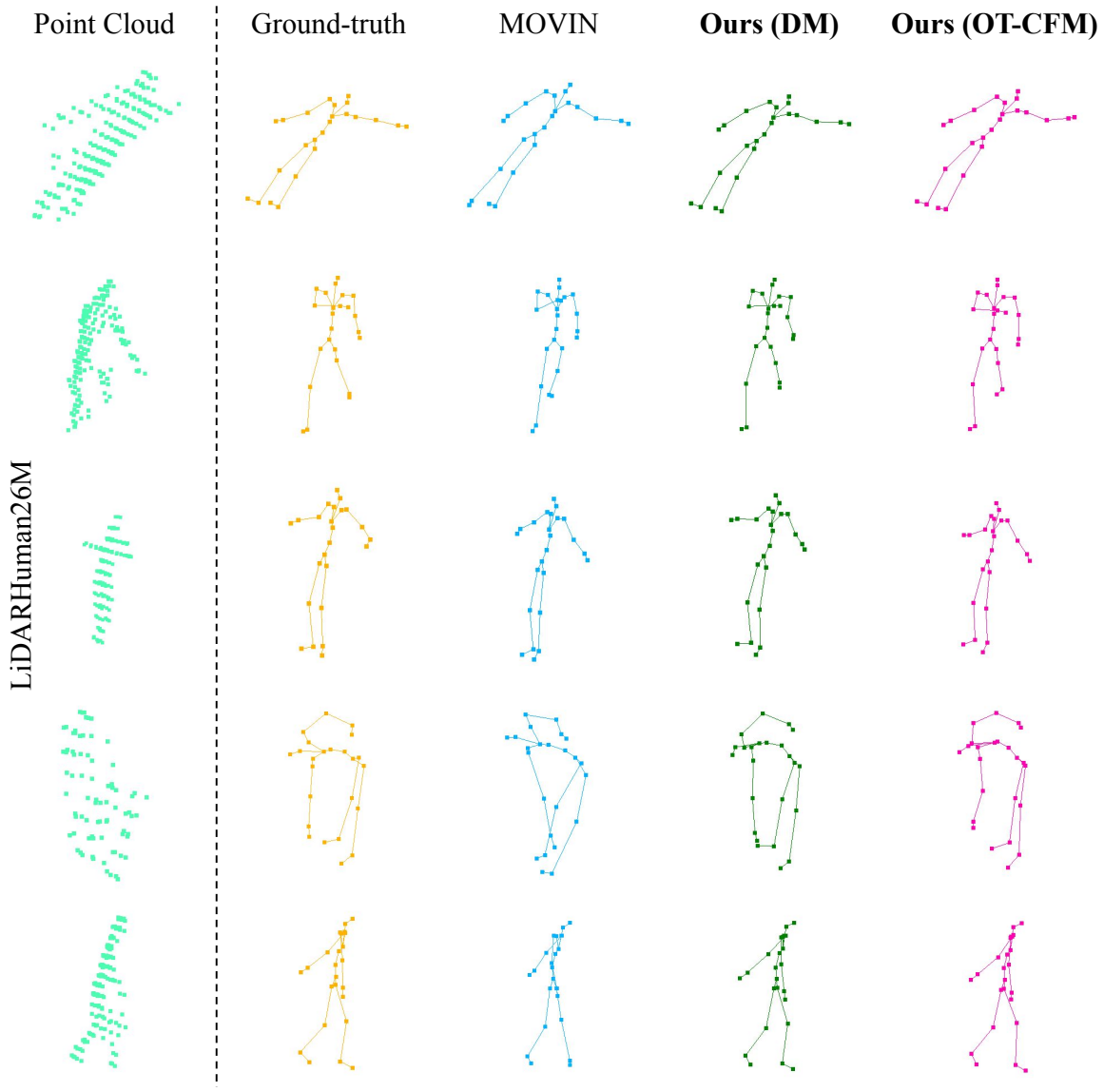

Figure 6. Qualitative results of pose estimation using the proposed method and MOVIN [3] on the LiDARHuman26M [4] dataset. For both Point2Pose and MOVIN, the ground truth pose is used as input to the model. Point2Pose demonstrates more accurate pose estimation compared to MOVIN.

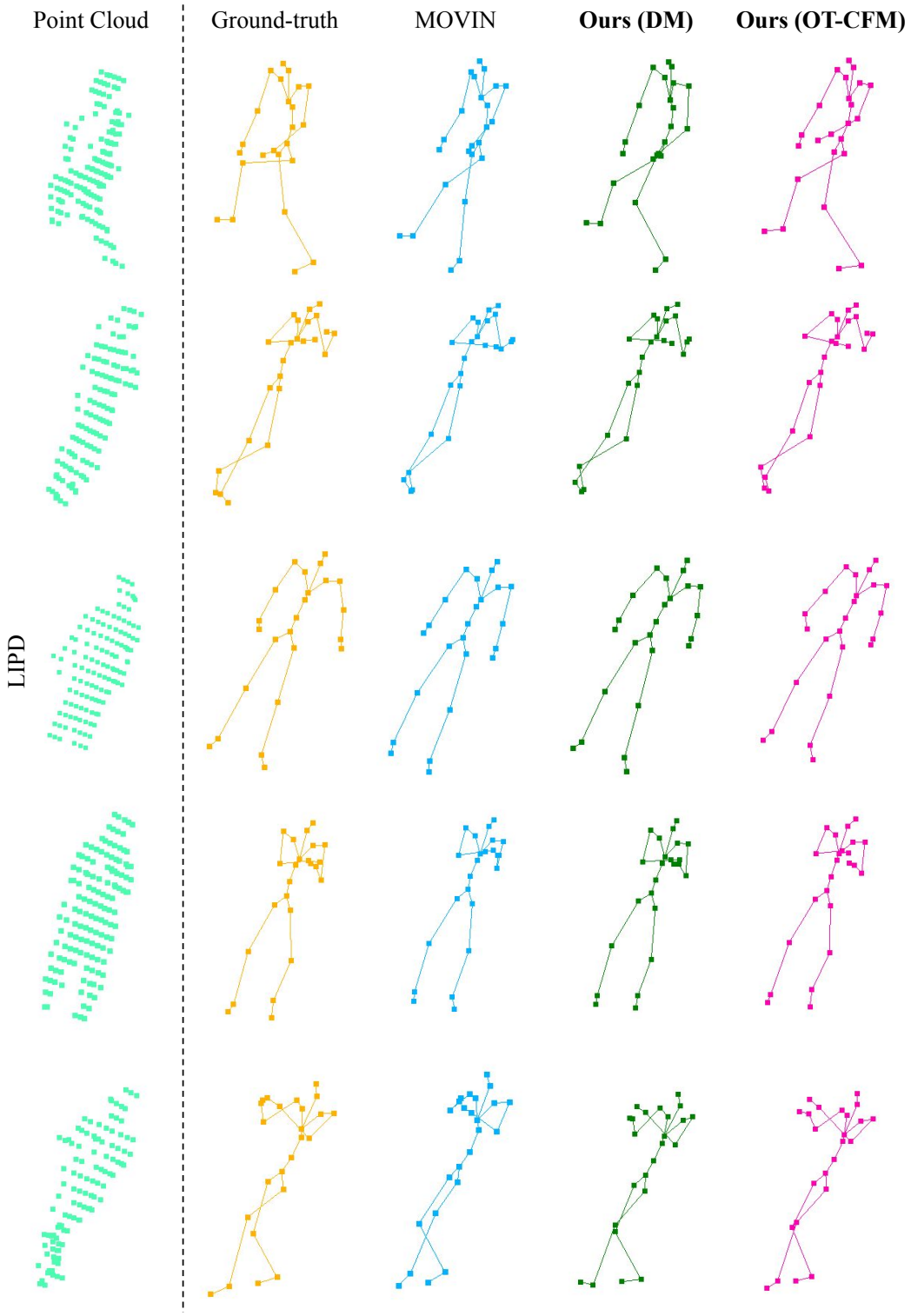

Figure 7. Qualitative results of pose estimation using the proposed method and MOVIN [3] on the LIPD [6] dataset. For both Point2Pose and MOVIN, the ground truth pose is used as input to the model. Point2Pose demonstrates more accurate pose estimation compared to MOVIN.

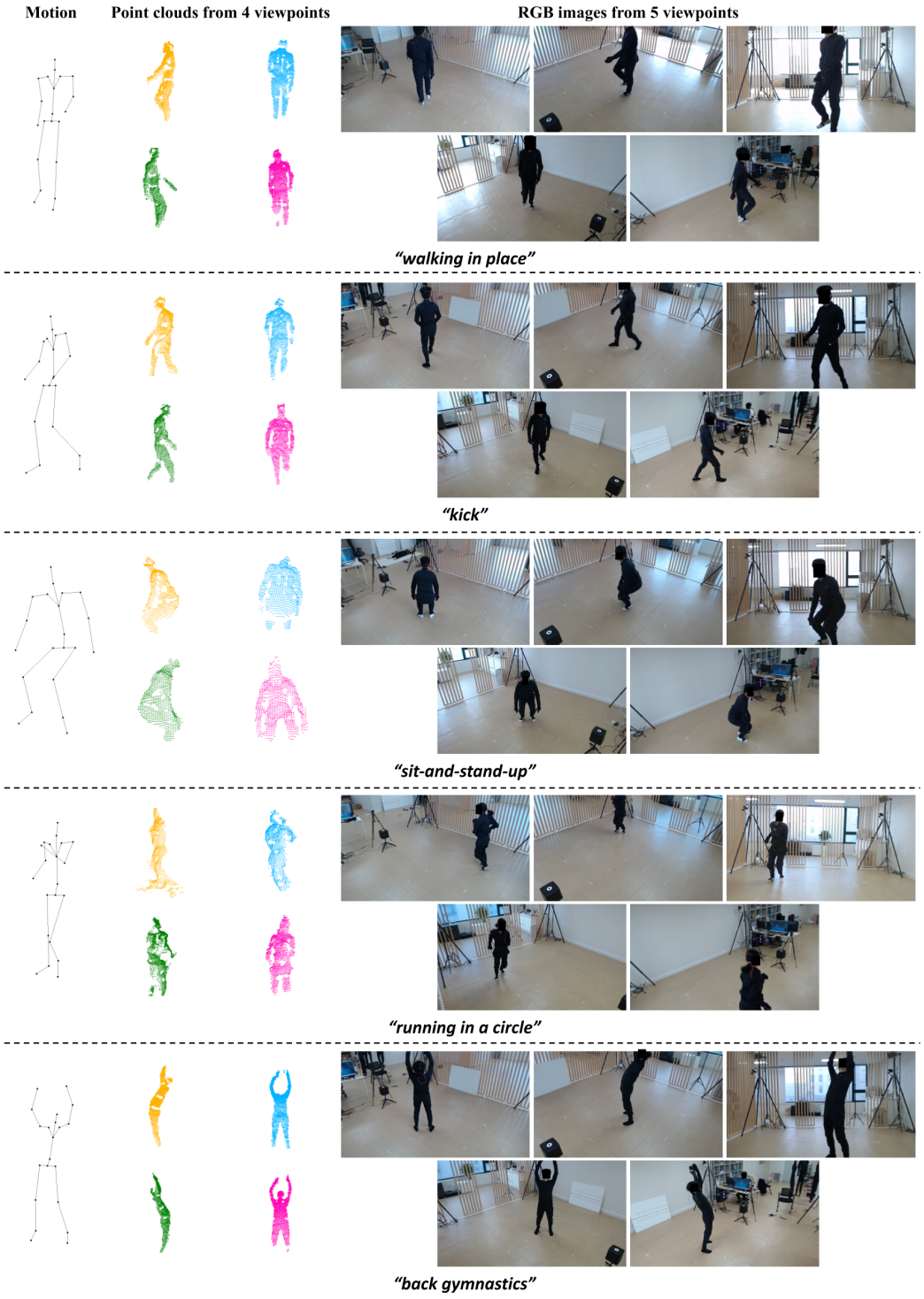

Figure 8. Samples from the proposed dataset. The MVPose3D dataset includes a diverse range of human motions. Each sample consists of four point clouds, five RGB images, 3D joint coordinates, and joint rotation data. We mask each participant's face for visualization.

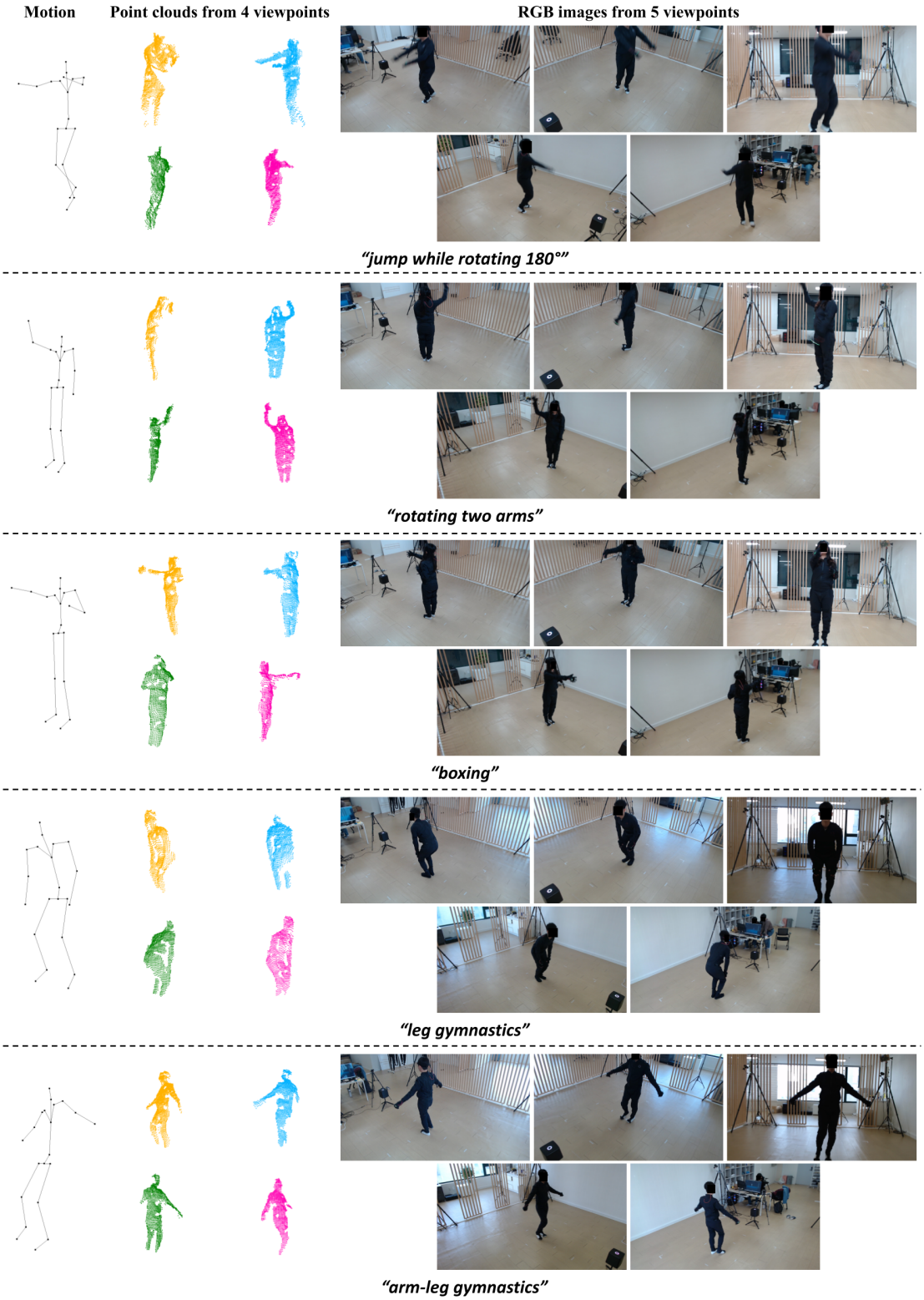

Figure 9. Samples from the proposed dataset. The MVPose3D dataset includes a diverse range of human motions. Each sample consists of four point clouds, five RGB images, 3D joint coordinates, and joint rotation data. We mask each participant's face for visualization.