Supplementary Materials

Anonymous Authors

1 DATASET DETAILS

PSU-TMM100 are measured by Vicon motion capture system, which has 12 IR cameras and 39 optical markers. In order to obtain the ground truth of the SMPL model parameters, we first process the original 3D marker points. As shown in Table 1, we match the marker index and the SMPL vertex index according to [5]. Then we utilize Mosh++ [5] to obtain SMPL parameters ground-truth.

2 ROBOT CONTROL

The whole-body control of the humanoid can be divided into two steps: the inverse kinematic (IK) solver and the stabilizer. The IK solver determines a joint configuration that enables the joints to reach the target positions, while the stabilizer ensures that the motion adheres to balance and stability constraints.

Firstly, given a target pose, we employ a differential IK approach, which can be denoted as

$$\begin{aligned} \min_{\dot{Q}} \quad & \sum_i \omega_i \| \mathcal{J}_i \dot{Q} - K_i \boldsymbol{v}_i \|_2^2 + \|\dot{Q}\|_2^2 \\ \text{s.t.} \quad & K_m \frac{Q_l - Q_c}{dt} \le \dot{Q} \le K_m \frac{Q_u - Q_c}{dt}. \end{aligned} \tag{1}$$

Here, $\mathcal{J}_i$ represents the Jacobian matrix, which maps joint velocities $\dot{Q}$ to the velocities of end-point i. $\boldsymbol{v}_i$ denotes the velocity residual, defined as the difference between the current and target positions of the end-point. By multiplying $\boldsymbol{v}_i$ by a proportional gain K_i, we ensure that $\mathcal{J}_i\dot{Q}$ endeavors to reach the desired values at each time step dt, allowing us to obtain the desired $\dot{Q}$. In practice, we can set implicit priorities between targets by having e.g. $\omega_i = 10^4$ for the most important end-point positions (such as elbows, wrists, etc.) to preserve motion similarity, and $\omega_i = 10^2$ to the subsequent ones (like ankles, etc.) to ensure stability [4]. Additionally, we consider $\|\dot{Q}\|_2^2$ to keep the $\dot{Q}$ as continuous as possible.

In this process, the $\dot{Q}$ must be constrained within the lower and upper limit of joint variation. In Eq. 1, Q_c represents the current joint angle, Q_l is the lower limit of the joint angle, and Q_u is the upper limit of the joint angle. K_m is a proportional gain which means a joint angle update will not exceed the gap separating its current value from its bounds.

We use QP solvers, CVXOPT [1] and OSQP [3] in Python to solve the problem described in Eq. 1. For mathematical derivation and more details, readers can refer to [2].

Having got the joint velocities $\dot{Q}$, we can update the joint configuration by

$$\hat{Q} = \dot{Q} + Q_c, \tag{2}$$

where $\hat{Q}$ is the target joint configuration. However, the robot can not directly be driven by $\hat{Q}$ due to the risk of losing balance. So we design a stabilizer to further optimize the joint configuration, which can be represented as

$$\begin{aligned} \min_{Q} \quad & \|\hat{Q} - Q\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{M}_R(Q) \subseteq C^m_{robot} \\ & \boldsymbol{J}_R^{lf,rf}(Q) \subseteq C^j_{robot} \\ & \boldsymbol{R}_R^{lf,rf}(Q) \subseteq C^r_{robot}. \end{aligned} \tag{3}$$

Here, $\boldsymbol{M}_R$ represents the CoM position of the robot, which must be constrained within the support polygons C^m_{robot} [6]. Additionally, $\boldsymbol{J}_R^{lf,rf}$ denotes the positions of the two feet, which need to be restricted in foot position constraints C^j_{robot} to ensure stability. For instance, when the humanoid is supported by both feet, the heights of both feet must be equal, when supported by the left foot, the height of the right foot must not be lower than that of the left foot. Similarly, the foot rotations $\boldsymbol{R}_R^{lf,rf}$ must also be restricted in rotation constraints C^r_{robot}. For example, when the humanoid is supported by both feet, the rotation of both feet should be consistent.

3 ADDITIONAL EXPERIMENTS

3.1 Similarity Visualization

We compare the trajectories of human(ground-truth) and robot joints in Fig 2. It's evident that the upper-body joint trajectories (i.e., Elbows and Wrists) generated by the our RGB-P method are consistent with those of the RGB method, indicating that our method doesn't alter the upper-body imitation similarity. However, there is a substantial decline in similarity observed in the lower-body joint trajectories (i.e., ankles), which occasionally differ from those of the RGB method. This discrepancy arises from the need to correct trajectories for stability and balance maintenance. We posit that this compromise in similarity is essential to ensure the robot's safety.

3.2 Stability Visualization

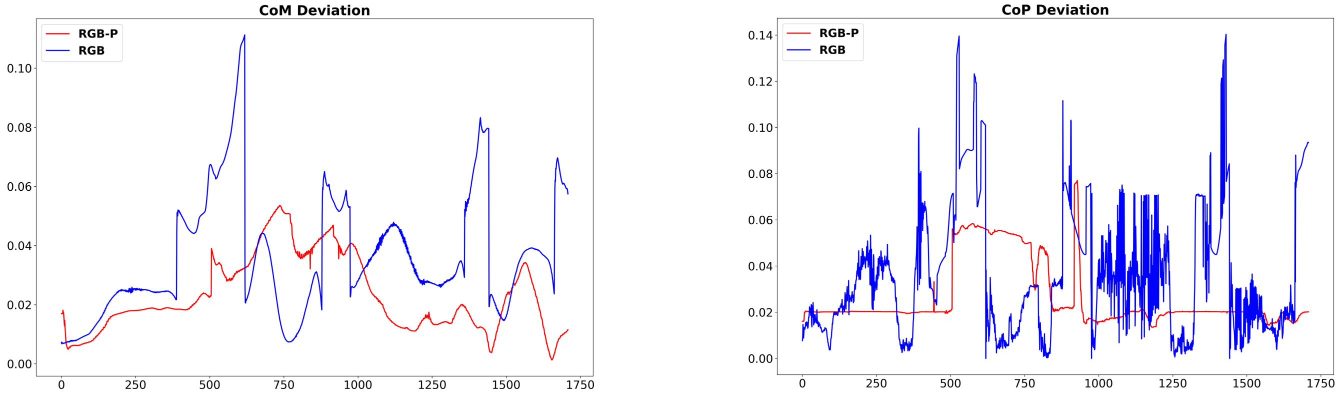
As depicted in Fig. 1, the curves generated by the RGB method display considerable jitter, especially in the case of CoP deviation, which directly detected by foot sensors. This jitter suggests instability in humanoid. In contrast, the curves from our RGB-P method appear smoother and more stable, demonstrating that the integration of pressure data into the RGB method substantially improves stability.

3.3 Imitation Video

We also provide a video showcasing the imitation process in both simulation and a real environment. In the simulation, the RGB method fails to adjust the CoM to match the support mode, resulting in a lean in the body. This is primarily due to the imprecise estimation of the human CoM by the RGB method. In contrast, our RGB-P method accurately captures the changes in the human CoM and maps them to the humanoid, successfully completing the motion of lifting the leg.

Table 1: Marker and vertex correspondence table

Marker Index	Marker Name	Vertex Index	Marker Index	Marker Name	Vertex Index
1	LFHD	0	21	RWRA	5573
2	RFHD	3512	22	RWRB	5568
3	LBHD	182	23	RFIN	5635
4	RBHD	3694	24	LASI	857
5	C7	3470	25	RASI	4343
6	T10	3016	26	LPSI	3122
7	CLAV	3171	27	RPSI	6544
8	STRN	3506	28	LTHI	1454
9	RBAK	5273	29	LKNE	1053
10	LSHO	1861	30	LTIB	1112
11	LUPA	1443	31	LANK	3327
12	LELB	1666	32	LHEE	3387
13	LFRM	1568	33	LTOE	3233
14	LWRA	2112	34	RTHI	4927
15	LWRB	2108	35	RKNE	4538
16	LFIN	2174	36	RTIB	4598
17	RSHO	5322	37	RANK	6728
18	RUPA	4918	38	RHEE	6786
19	RELB	5135	39	RTOE	6633
20	RFRM	5037			



(a) CoM Deviation

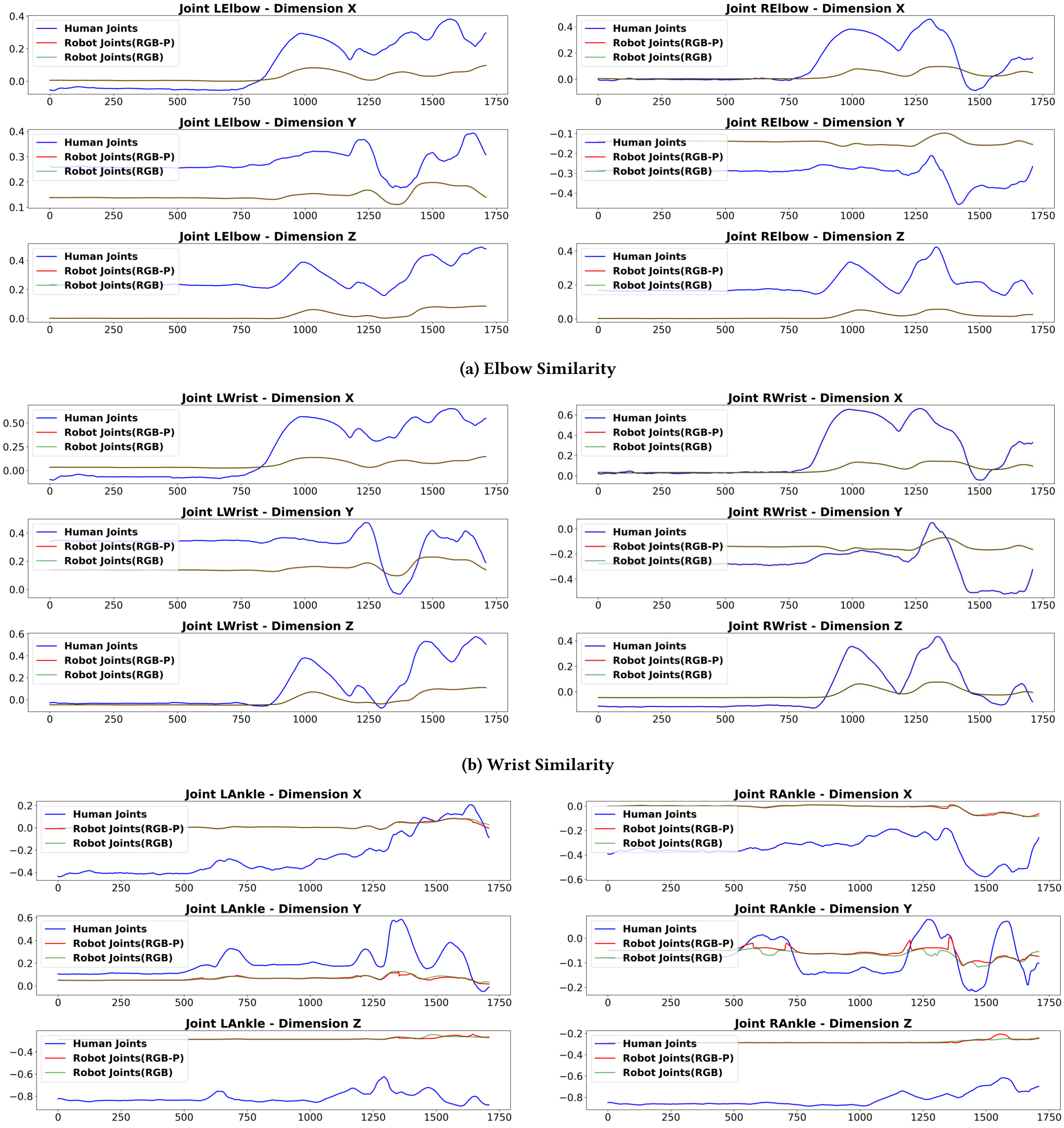
(b) CoP Deviation

Figure 1: Comparison of CoM and CoP deviations.

Additionally, we conduct an experiment with a real robot performing a series of Taiji motions learned from a human. Due to the low efficiency of the robot's servomotors, we have accelerated the humanoid motion video to display it conveniently at the same pace as the human motion. Comparing the robot's motion with that of a human, we can observe that the humanoid moved in a similar trend while maintaining its balance and stability, which further confirms the effectiveness of our method.

REFERENCES

[1] 2024. *CVXOPT.* https://cvxopt.org/.
[2] 2024. *Inverse kinematics.* https://scaron.info/robotics/inverse-kinematics.html.
[3] 2024. *OSQP.* https://osqp.org/docs/get_started/.
[4] Stéphane Caron and Abderrahmane Kheddar. 2016. Multi-contact walking pattern generation based on model preview control of 3D COM accelerations. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 550–557.
[5] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5442–5451.
[6] Eiichi Yoshida, Oussama Kanoun, Claudia Esteves, and Jean-Paul Laumond. 2006. Task-driven support polygon reshaping for humanoids. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 208–213.



(a) Elbow Similarity

(b) Wrist Similarity

(c) Ankle Similarity

Figure 2: Comparison of similarities in different joints.