Appendix
DiffusionPoser: Real-time Human Motion Reconstruction From Arbitrary Sparse Sensors Using Autoregressive Diffusion

A. DiffusionPoser for OpenSim

We implemented DiffusionPoser for another skeleton model, the OpenSim skeleton. This model has been used in hundreds of biomechanical studies [12]. In addition to having physiologically-realistic joint definitions, OpenSim models the musculoskeletal system providing information such as muscle-tendon lengths, which can be used for computation of muscle forces, and energy consumption as in [1, 2].

A.1. Skeleton and motion representation

We used the Lai musculoskeletal model [6] from OpenSim (Figure 1). This model comprises 31 degrees of freedom (pelvis [6], hips [3], knees [1], ankles [1], subtalar joints [1], lumbar [3], shoulders [3], elbows [1], radioulnar joints[1]). The translational and orientation offsets between different joints are physiologically realistic. For example, the knee joint rotation axis translates and rotates as a function of the knee angle. Because the torso is rigid and there is no neck joint, we do not allow an IMU placed at the head for this implementation. As such, the OpenSim model has fewer degrees of freedom and one less IMU than SMPL, which has implications for the motion representation. The motion is now represented by $\boldsymbol{x} \in \mathbb{R}^{61\times135=61\times(16\times6+12\times3+2+1+4)}$ to represent 61 frames of 16 joint orientations (6dof representation), 12 IMU accelerations of bone specific sites (3dof), horizontal position change (2dof), vertical position (1dof) and 4 contact labels.

A.2. Training Data

Since no large dataset of motion synthesized with OpenSim models exists, we created one from existing marker-based optical motion capture datasets. We synthesized motions and associated musculoskeletal models scaled to subject anthropometry. To this end, we used AddBiomechanics, a new tool that automates the estimation of joint kinematics from marker data [13]. We used marker-based optical motion capture data from three existing datasets [5, 10, 11] to synthesize motions. The aggregated dataset comprises data from 339 subjects and 3350 trials. Synthesizing training data is identical to the procedure for the SMPL implementation.

A.3. Quantitative Results

We compare selected configurationurations from DiffusionPoser for OpenSim against TIP [4]. Due to the differences in the OpenSim skeleton and SMPL, we need to establish joint correspondence when calculating MJOE and MJPE. For each joint in OpenSim, we found the closest corresponding joint in SMPL. Joints on the limbs can be exactly mapped between skeletons, but SMPL has many more joints proximal to the root which are not all included in the metric calculation. Although the mapping for these proximal joints is not exact between skeletons, we found that proximal joints nevertheless have low position error compared to joints on the limbs, as the error increases when progressing along the kinematic tree.
We show evaluation metrics for TIP [4] and DiffusionPoser for OpenSim in Table 1. We show results for DiffusionPoser with four different configurationurations with six, four and three sensors. Similar to DiffusionPoser for SMPL we found that DiffusionPoser for OpenSim is close to TIP for different evaluation metrics. The degradation of the evaluation metrics with more sparse sensor configurationurations is similar for the OpenSim and SMPL version as well.

Table 1. **Comparison between selected configurationurations of DiffusionPoser for six, four and three sensors and TIP on the TotalCaptureReal dataset.**

configuration	LA[°]	JPE[cm]	RE2[m]	RE10[m]
TIP	12.1	6.8	0.17	0.32
$plvs, torso, ft_r, ft_l, wr_r, wr_l$	13.6	6.3	0.14	0.35
ft_r, ft_l, wr_r, wr_l	14.7	7.8	0.14	0.37
$plvs, ft_r, ft_l$	19.7	11.9	0.19	0.50
$plvs, ft_r, wr_l$	18.6	10.8	0.22	0.82

A.4. Ablation with conditional transformer with sensor configuration mask

We compared performance of DiffusionPoser for OpenSim, which is a generative model against a regression model using a conditional transformer. We chose the same transformer architecture for the conditional transformer as for DiffusionPoser and concatenate a mask that represents the IMU configuration to the input features. During training we randomly sample across IMU configurations and mask out the non-measured parts of the feature vector for the final frame. The first 60 frames are assumed given as input during training, where we dropout 80% of the non-measured features to avoid overfitting [4, 9]. As we will deploy this transformer in an online autoregressive fashion we only care about predicting the last frame. At inference, we shift the latest prediction into the history at every new frame.

DiffusionPoser performs better than the conditional transformer across different IMU configurations (Table 2). From analyzing the conditional transformer predictions we found that for some trials it is on par with DiffusionPoser while it fails in other cases. We attribute robustness of DiffusionPoser to its generative nature.

Table 2. **Evaluation of a conditional transformer fulfilling the task of DiffusionPoser.**

configuration	system	LA[°]	JPE[cm]	RE10[m]
$plvs, torso, ft_r, ft_l, wr_r, wr_l$	conditional	19.1	11.8	1.15
	generative	**13.6**	**6.3**	**0.35**
ft_r, ft_l, wr_r, wr_l	conditional	22.5	14.7	1.47
	generative	**14.7**	**7.8**	**0.37**
$plvs, ft_r, ft_l$	conditional	25.9	18.1	1.58
	generative	**19.7**	**11.9**	**0.50**

A.5. Qualitative Results

A video with results from DiffusionPoser for OpenSim is to be found on our project website.

B. Baseline comparisons for six IMU sensors

B.1. Details on evaluation metrics

We compared DiffusionPoser to state-of-the-art methods that focus on reconstructing motion from a specific six IMU sensor configuration: Transpose [15], TIP [4] and PIP [14]. We evaluated these three systems and DiffusionPoser on the same evaluation metrics and provide a script that unifies the evaluation code and metrics from the prior work for a fair and direct comparison. Our evaluation metrics for Transpose, TIP and PIP have slightly different numerical values than those reported in the respective papers for corresponding metrics. This has several reasons:

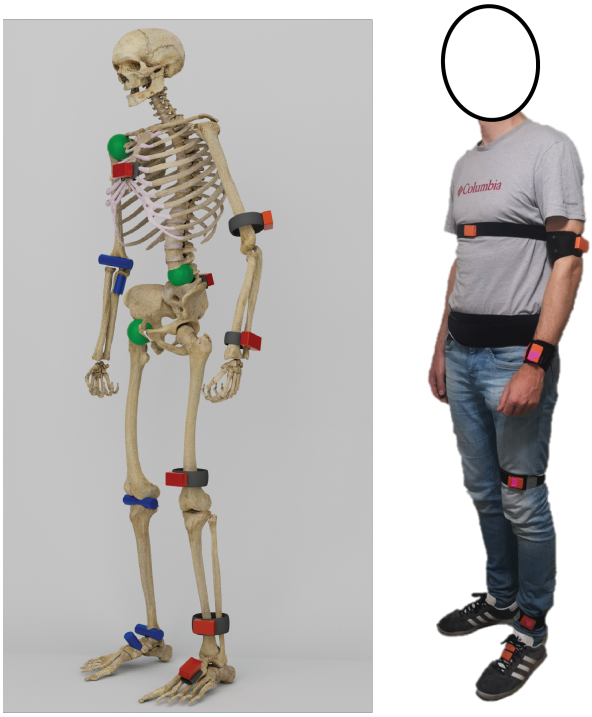

Figure 1. (Left) **OpenSim kinematic model with IMUs**. Our model includes 10 hinge joints shown in blue and 5 ball-and-socket joints shown in green, resulting in 31 degrees of freedom. IMU location canditates are shown in orange. (Right) **User instrumented with IMUs**.

- For PIP and Transpose, the ground truth for TotalCapture trials comes from the DIP paper [3] and is the result of reconstructing motion from using marker information. TIP however uses the published ground truth poses from the TotalCapture split in AMASS. These reconstructions are also based on marker information, but are slightly different. Here we used the DIP reconstructions of TotalCapture for all evaluations.
- PIP and Transpose evaluated on more TotalCapture trials than TIP. The reason is that the ground truth for TIP (see above) does not have reconstructions for all trials. The ground truth provided by DIP and used by Transpose and PIP does have all reconstructions, and this is why we used this split for evaluation.
- For global angular errors there is some inconsistency as to whether the root orientations are first aligned before calculating the errors. We first aligned the root orientations between reconstruction and ground truth before calculating joint position errors and global orientation errors.
- For the global angular errors, the local angular errors and the joint position errors there was no consistent set of joints across which values were averaged. Because

Table 3. **Comparison of DiffusionPoser to different baselines for DIP-IMU**. Numbers are averages over all trials, bracketed numbers are the metrics from the trial with the highest error.

system	LA [°]	GA [°]	JPE [cm]
Transpose	13.1(18.5)	12.3(17.2)	6.1(9.5)
PIP	12.4(20.1)	12.4(19.2)	5.2(8.6)
TIP	12.4(28.1)	11.7(21.0)	5.7(13.4)
DiffusionPoser	12.4(18.8)	11.6(19.4)	5.3(9.2)

some joints are ignored, f.e. the root, wrists, fingers, toes and ankles (only by PIP and Transpose), these errors were set to zero. However, not all of these zeroed values were excluded from averaging which leads to lower average errors. For the joint error and global orientation error we ignored the root joint (which has zero error due to alignment) and the wrist, finger and toe joints because this degree of freedom was not estimated.

B.2. Comparison on DIP-IMU dataset

Besides comparing DiffusionPoser against other systems on TotalCaptureReal (see main paper), we also evaluated on DIP-IMU [3], which has real IMU data coupled with ground truth motion as well. Similar to Transpose, PIP and TIP we finetuned on the first eight subjects of the DIP dataset and tested on subject nine and ten. Since DIP-IMU does not have ground truth for root motion, we first used the model before finetuning to get a reasonable reconstruction of the ground truth root motion to complete the training data. Results form the comparisons are in Table 3. Similar to TotalCapture, results are very close between the four systems.

C. Evaluation results for different IMU configurations

In this section we provide tables with detailed results for all the configurations for which we tested DiffusionPoser.

We start with a summary graph that compares accuracy on different metrics for the best configuration give a number of IMUs (figure 2).

Table 4 shows results for the different configurations we tested on TotalCaptureSynth. We tested configurations with four, three and two sensors. For the case of three and four IMUs we limited ourselves to cases with symmetry for instrumented limbs (e.g. if there is a sensor on the right foot, there is one on the left foot as well). Conclusions on these results are outlined in the main paper. Table 5 reports the same metrics but for all configurations tested (between one and six sensors) on TotalCaptureReal. Here we are more limited in possible configurations because we only have IMU data for head, pelvis, wrists and shanks.

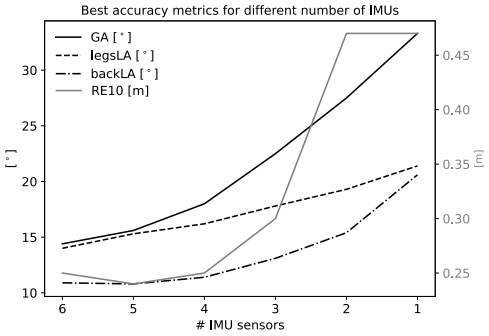

Figure 2. Accuracy on different metrics of the best configuration for a given number of IMUs. Tested for 1 to 6 IMUs that could be attached to wrists, shanks, pelvis and head.

Table 4. **Evaluation of different configurations using four, three and two sensors on 'TotalCaptureSynth'.** Darker cell colors indicate better metrics. Colors are normalized per metric and per number of sensors.

configuration	GA [°]	legsLA[°]	backLA[°]	Jitter[-]	RE2[m]	RE10[m]
$plvs, head, th_r, th_l$	22.3	7.0	13.0	2.9	0.11	0.31
$plvs, head, sh_r, sh_l$	22.3	11.1	13.1	2.5	0.08	0.17
$plvs, head, ft_r, ft_l$	21.8	12.1	13.4	3.0	0.12	0.23
$plvs, head, arm_r, arm_l$	14.4	15.6	7.6	3.4	0.18	0.74
$plvs, head, wr_r, wr_l$	11.8	15.1	8.4	3.3	0.40	0.76
th_r, th_l, sh_r, sh_l	15.0	7.5	14.0	2.0	0.10	0.11
th_r, th_l, ft_r, ft_l	24.4	10.6	13.9	2.8	0.09	0.14
th_r, th_l, arm_r, arm_l	13.8	9.7	8.2	3.1	0.20	0.30
th_r, th_l, wr_r, wr_l	13.1	10.8	9.3	3.1	0.20	0.33
sh_r, sh_l, ft_r, ft_l	26.7	9.5	14.9	2.9	0.15	0.24
sh_r, sh_l, arm_r, arm_l	16.6	15.2	9.7	2.9	0.13	0.20
sh_r, sh_l, wr_r, wr_l	16.0	15.9	11.7	2.9	0.14	0.22
ft_r, ft_l, arm_r, arm_l	15.1	16.2	11.0	3.2	0.08	0.23
ft_r, ft_l, wr_r, wr_l	14.9	16.4	11.2	3.3	0.08	0.23
arm_r, arm_l, wr_r, wr_l	19.3	11.3	18.3	3.2	0.24	1.20
$plvs, th_r, th_l$	27.8	7.5	14.0	2.9	0.14	0.40
$plvs, sh_r, sh_l$	28.1	12.7	14.2	2.5	0.10	0.21
$plvs, ft_r, ft_l$	27.7	13.7	14.5	2.9	0.09	0.25
$plvs, arm_r, arm_l$	17.9	19.3	8.9	3.4	0.20	0.90
$plvs, wr_r, wr_l$	17.2	20.9	10.2	3.5	0.21	0.76
hd, arm_r, arm_l	19.8	16.5	10.2	3.1	0.26	1.03
hd, th_r, th_l	23.1	10.1	13.6	3.0	0.16	0.37
hd, sh_r, sh_l	19.8	16.5	10.2	3.1	0.26	1.03
hd, ft_r, ft_l	19.8	16.5	10.2	3.1	0.26	1.03
hd, wr_r, wr_l	18.4	16.1	11.2	3.2	0.23	0.95
$plvs, th_r$	32.5	12.0	14.7	2.9	0.17	0.75
$plvs, th_l$	34.0	12.4	15.4	3.4	0.20	0.66
$plvs, sh_r$	32.9	15.2	15.0	2.8	0.13	0.50
$plvs, sh_l$	23.5	15.1	13.1	3.4	0.16	0.56
$plvs, ft_r$	32.3	15.5	14.8	2.9	0.16	0.48
$plvs, ft_l$	26.4	16.5	16.3	3.9	0.18	0.52
th_r, th_l	27.1	13.9	14.9	2.9	0.15	0.59
sh_r, sh_l	29.9	17.0	16.4	3.1	0.12	0.26
ft_r, ft_l	28.2	15.5	15.6	3.0	0.09	0.34
$plvs, hd$	31.6	18.4	15.2	3.1	0.25	1.04
$plvs, arm_r$	23.8	17.6	13.5	3.1	0.26	1.06
$plvs, arm_l$	23.1	19.5	13.2	2.7	0.25	0.94
$plvs, wr_r$	23.0	18.0	14.0	2.3	0.26	1.26
$plvs, wr_l$	23.1	18.9	13.9	3.0	0.30	1.13
hd, arm_r	23.5	17.2	13.2	3.0	0.28	1.18
hd, arm_l	23.2	19.4	13.0	2.8	0.28	0.84
hd, wr_r	23.0	18.6	14.0	3.3	0.24	1.16
hd, wr_l	23.1	19.1	13.9	2.8	0.27	0.88
arm_r, arm_l	22.9	20.8	12.7	3.3	0.28	1.17
wr_r, wr_l	26.6	22.9	16.5	3.1	0.32	1.00

Table 5. **Evaluation of different configurations using six, five, four, three, two and one sensors on 'TotalCaptureReal'.** We test all possible with six or fewer IMUs across six potential attachment sites: pelvis, head, wrists, shanks. Darker cell colors indicate better metrics. Colors are normalized per metric and per number of sensors.

configuration	GA [°]	legsLA[°]	backLA[°]	Jitter[-]	RE2[m]	RE10[m]
$plvs, hd, wr_r, wr_l, sh_r, sh_l$	14.4	14.0	10.9	2.8	0.14	0.25
$plvs, wr_r, wr_l, sh_r, sh_l$	18.0	16.9	12.5	2.8	0.14	0.26
$plvs, hd, wr_r, sh_r, sh_l$	19.4	14.9	12.1	2.7	0.13	0.24
$plvs, hd, wr_l, sh_r, sh_l$	19.9	15.3	13.0	3.1	0.14	0.27
$plvs, hd, wr_r, wr_l, sh_r$	15.9	15.8	10.8	2.9	0.24	0.50
$plvs, hd, wr_r, wr_l, sh_l$	15.6	15.7	10.9	2.9	0.25	0.46
$hd, wr_r, wr_l, sh_r, sh_l$	17.7	17.2	12.4	2.9	0.16	0.29
$plvs, wr_r, sh_r, sh_l$	24.2	16.2	13.4	2.7	0.12	0.26
$plvs, wr_l, sh_r, sh_l$	24.9	18.0	14.1	3.1	0.13	0.25
$plvs, hd, sh_r, sh_l$	29.4	15.8	16.9	2.7	0.14	0.33
$plvs, wr_r, wr_l, sh_r$	20.1	18.7	12.8	2.9	0.26	0.51
$plvs, hd, wr_r, sh_r$	21.1	16.7	12.3	2.8	0.23	0.40
$plvs, hd, wr_l, sh_r$	22.5	17.1	13.6	3.2	0.26	0.45
$plvs, wr_r, wr_l, sh_l$	19.3	18.5	12.7	3.0	0.28	0.55
$plvs, hd, wr_r, sh_l$	20.6	16.8	12.4	2.8	0.22	0.48
$plvs, hd, wr_l, sh_l$	22.1	18.1	14.1	3.1	0.26	0.49
$plvs, hd, wr_r, wr_l$	18.0	19.2	11.4	3.2	0.34	0.96
wr_r, wr_l, sh_r, sh_l	21.6	20.4	15.1	3.1	0.19	0.33
hd, wr_r, sh_r, sh_l	24.3	19.5	15.2	2.8	0.16	0.33
hd, wr_l, sh_r, sh_l	22.1	18.7	14.1	3.2	0.17	0.33
hd, wr_r, wr_l, sh_r	21.1	19.1	13.9	2.9	0.28	0.57
hd, wr_r, wr_l, sh_l	19.8	19.4	13.2	3.0	0.29	0.60
$plvs, sh_r, sh_l$	36.4	17.8	18.0	2.8	0.13	0.30
$plvs, sh_r, wr_r$	27.5	19.1	14.2	2.9	0.23	0.47
$plvs, sh_r, wr_l$	29.0	20.2	14.9	3.4	0.24	0.60
$plvs, sh_r, hd$	31.0	18.4	16.9	2.7	0.25	0.50
$plvs, sh_l, wr_r$	26.5	17.9	13.8	2.8	0.26	0.49
$plvs, sh_l, wr_l$	27.9	20.7	15.3	3.1	0.27	0.55
$plvs, sh_l, hd$	29.9	18.3	16.4	3.1	0.26	0.51
$plvs, wr_r, wr_l$	22.5	22.9	13.1	3.3	0.31	1.11
$plvs, wr_r, hd$	24.0	22.2	13.3	3.1	0.35	1.24
$plvs, wr_l, hd$	24.7	21.5	13.5	3.3	0.39	1.10
sh_r, sh_l, wr_r	26.4	19.1	15.5	3.1	0.18	0.40
sh_r, sh_l, wr_l	25.0	19.8	15.2	3.8	0.19	0.36
sh_r, sh_l, hd	31.4	17.9	18.3	2.9	0.17	0.39
sh_r, wr_r, wr_l	25.0	22.6	16.5	3.1	0.31	0.72
sh_r, wr_r, hd	24.9	19.8	15.2	2.8	0.26	0.71
sh_r, wr_l, hd	25.3	20.8	15.9	3.1	0.31	0.60
sh_l, wr_r, wr_l	24.9	22.5	16.5	3.4	0.33	0.65
sh_l, wr_r, hd	25.0	21.0	15.3	2.9	0.31	0.60
sh_l, wr_l, hd	23.5	20.0	14.7	3.2	0.35	0.64
wr_r, wr_l, hd	24.0	21.0	14.7	3.0	0.39	1.16
$plvs, sh_r$	40.6	20.6	18.7	2.8	0.26	0.72
$plvs, sh_l$	41.5	21.5	18.7	3.1	0.28	0.61
$plvs, wr_r$	33.6	24.7	15.4	3.8	0.35	1.35
$plvs, wr_l$	35.9	25.4	16.4	3.9	0.37	1.27
$plvs, hd$	33.3	21.4	16.8	2.5	0.33	1.32
sh_r, sh_l	39.2	24.6	20.1	4.3	0.21	0.47
sh_r, wr_r	28.4	21.5	16.9	3.1	0.29	0.83
sh_r, wr_l	30.7	24.9	19.0	3.3	0.35	0.83
sh_r, hd	32.9	19.3	19.0	2.9	0.28	0.65
sh_l, wr_r	31.1	24.4	18.2	3.9	0.45	0.85
sh_l, wr_l	27.8	22.4	16.8	3.7	0.38	0.79
sh_l, hd	33.3	22.1	19.4	3.7	0.36	0.67
sh_r, wr_l	32.1	25.6	20.7	3.3	0.47	1.47
wr_r, wr_l	27.5	21.7	16.9	2.7	0.40	1.37
wr_l, hd	28.2	23.1	17.2	2.9	0.47	1.16
$plvs$	40.6	29.6	20.6	2.8	0.26	0.72
sh_r	41.5	26.1	22.1	3.1	0.28	0.61
sh_l	33.6	28.0	21.7	3.8	0.35	1.35
wr_r	35.9	21.8	21.8	3.9	0.37	1.27
wr_l	33.3	26.3	24.4	2.4	0.33	1.32
$head$	39.2	21.4	20.6	4.3	0.21	0.47

Finally, in Table 6, we provide additional comparisons between using synthetic and real IMU data as input. We showed and discussed selected results in the main paper (Table 2). As explained in the main paper the imperfect sensor-to-bone callibration and relative motion between sensor and bone during motion are likely the largest contributors to explaining sim-to-real differences. To give an idea about this error we calculated the average global angular error between the sensor orientation estimation and the ground truth orientation of the bone to which the sensor is attached for the TotalCapture dataset. For the synthetic case this error is off course 0 degrees, but we found it to be 9.4 degrees for the real IMU data. Although this is not a strict upper bound for GA, as the model could learn to ignore extreme positions that would results from wrong callibration or a moving sensor, it is still important to understand such context when interpreting the errors.

Table 6. **Sim-to-real error for selected IMU configurations.**

configuration	IMU	GA [°]	legsLA[°]	backLA[°]	Jitter[-]	RE2[m]	RE10[m]
$plvs, hd, wr_r, wr_l, sh_r, sh_l$	real	14.4	14.0	10.9	2.8	0.14	0.25
	synthetic	7.0	9.0	7.3	2.8	0.09	0.17
$plvs, hd, wr_r, sh_r, sh_l$	real	19.4	14.9	12.1	2.7	0.13	0.25
	synthetic	10.7	10.2	8.8	2.7	0.10	0.17
$plvs, hd, sh_r, sh_l$	real	24.9	15.8	13.4	2.7	0.12	0.26
	synthetic	22.3	11.1	13.1	2.5	0.08	0.17
$plvs, sh_r, sh_l$	real	36.4	17.8	18.0	2.8	0.13	0.30
	synthetic	28.1	12.7	14.2	2.5	0.10	0.21
sh_r, sh_l	real	39.2	24.6	20.1	4.3	0.21	0.47
	synthetic	29.9	17.0	16.4	3.1	0.12	0.26
$head$	real	39.2	21.4	20.6	4.3	0.21	0.47
	synthetic	29.9	17.0	16.4	3.1	0.12	0.26

D. Additional details

Masses assigned to different SMPL segments are reported in 7.

Table 7. Mass distribution for energy metric calculation. Segment masses are loosely based on OpenSim2392 [8]

segment	mass[kg]
pelvis	11.7
thigh	9.3
spine_1	7.69
spine_2	3.84
spine_3	3.84
spine_4	1.92
collar	1.92
neck+head	1.92
upper arm	2.0
lower arm	1.2
hand	0.35
finger	0.1

Several hyperparameters can be found in Table 8.

Table 8. Hyperparameters for training

Hyperparameter	Value
Optimizer	AdamW [7]
Learning rate	1e-4
Batch size	256
Training steps	200K
# parameters	34M
Motion duration	3s
Frames per second	20
Diffusion steps	1000
β schedule	cosine

References

[1] Allison S. Arnold, May Q. Liu, Michael H. Schwartz, Sylvia Õunpuu, and Scott L. Delp. The role of estimating muscle-tendon lengths and velocities of the hamstrings in the evaluation and treatment of crouch gait. *Gait and Posture*, 23(3):273–281, 2006. 1

[2] Christopher L. Dembia, Nicholas A. Bianco, Antoine Falisse, Jennifer L. Hicks, and Scott L. Delp. Opensim moco: Musculoskeletal optimal control. *PLOS Computational Biology*, 16(12):1–21, 12 2021. 1

[3] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37:185:1–185:15, Nov. 2018. Two first authors contributed equally. 2, 3

[4] Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W. Winkler, and C. Karen Liu. Transformer inertial poser: Real-time human motion reconstruction from sparse imus with simultaneous terrain generation. In *SIGGRAPH Asia 2022 Conference Papers*, SA '22, New York, NY, USA, 2022. Association for Computing Machinery. 1, 2

[5] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015. 1

[6] Adrian KM Lai, Allison S Arnold, and James M Wakeling. Why are antagonist muscles co-activated in my simulation? a musculoskeletal model for analysing human locomotor tasks. *Annals of biomedical engineering*, 45:2762–2774, 2017. 1

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[8] Ajay Seth, Jennifer L Hicks, Thomas K Uchida, Ayman Habib, Christopher L Dembia, James J Dunne, Carmichael F Ong, Matthew S DeMers, Apoorva Rajagopal, Matthew Millard, et al. Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology*, 14(7):e1006223, 2018. 7

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2

[10] Agnieszka Szczesna, Monika Błaszczyszyn, and Magdalena Pawlyta. Optical motion capture dataset of selected techniques in beginner and advanced kyokushin karate athletes. *Scientific Data*, 8(1):13, 2021. 1

[11] Matt Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *2017 British Machine Vision Conference (BMVC)*, 2017. 1

[12] Scott D. Uhlrich, Thomas K. Uchida, Marissa R. Lee, and Scott L. Delp. Ten steps to becoming a musculoskeletal simulation expert: A half-century of progress and outlook for the future. *Journal of Biomechanics*, 154:111623, 2023. 1

[13] Keenon Werling, Michael Raitor, Jon Stingel, Jennifer L Hicks, Steve Collins, Scott Delp, and C Karen Liu. Rapid bilevel optimization to concurrently solve musculoskeletal scaling, marker registration, and inverse kinematic problems for human motion reconstruction. *bioRxiv*, pages 2022–08, 2022. 1

[14] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2

[15] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Trans. Graph.*, 40(4), jul 2021. 2