

SUPPLEMENTARY MATERIAL

This supplementary document provides additional information about QuaMo. Sec. A contains the details about the two neural networks used in the project, InitNet and ControlNet. Sec. B provides additional qualitative results with corresponding 2D projections of QuaMo’s predictions on the original images. Sec. C contains details about the training samples for the experiments on AIST dataset. Sec. D presents the computational complexity of QuaMo, and Sec. E discusses the approximated quaternion integration from prior work.

A NETWORKS DESIGN

The overall design of ControlNet is shown in Fig. 5. At every time step t , the ControlNet takes in as inputs the current human pose $q_t \in \mathbb{R}^{24 \times 4}$, angular velocity $\omega_t \in \mathbb{R}^{24 \times 3}$, reference pose $\hat{q}_t \in \mathbb{R}^{24 \times 4}$, root translation $r_t \in \mathbb{R}^3$, root velocity $v_t \in \mathbb{R}^3$, and reference root translation $\hat{r}_t \in \mathbb{R}^3$. All variables are concatenated into a single input vector of shape \mathbb{R}^{273} . The input is passed through two blocks of a sequential module, including a linear projection to embed dimension of 512, followed by LayerNorm and LeakyReLU activation. The output embedding with shape \mathbb{R}^{512} is then linearly projected to respective components of the meta-PD controller with acceleration enhancement.

Since neural networks work more stably with small numbers, we scale up the prediction of the parameters $\kappa_A, \kappa_P, \kappa_D$ by s_A, s_P, s_D respectively. The sigmoid functions ensure that the prediction is positive and within the range of $[0, s]$, leading to correct PD calculation and a stable ODE solving process, especially at the beginning of training. The chosen values for the scales are $s_A = 40, s_P = 40, s_D = 30$. For root translation, the scales are $s_P = 200, s_D = 200$. We found that these values are sufficient to prevent the ODE solver instability during training.

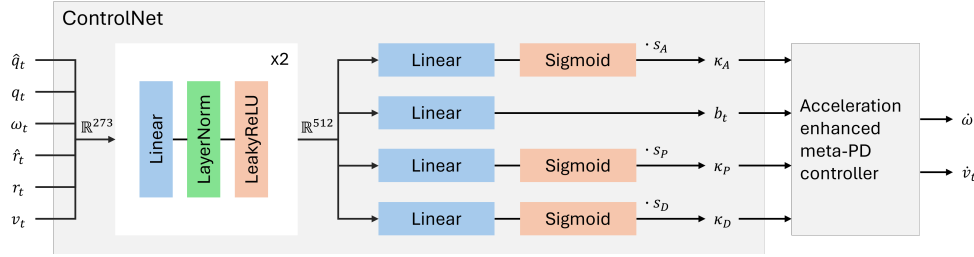


Figure 5: The architecture of ControlNet. The inputs are concatenated and linearly mapped to an embedding vector of 512 dimensions. The control parameters for the PD controller is predicted from the embedding via linear mappings, sigmoid functions and scaling.

To generate the initial solution for the QDE solvers, we implement an additional InitNet that takes in as input the first two reference poses $\hat{q}_{0:1}$, two reference root translations $\hat{r}_{0:1}$, and shape β_0 estimated from either TRACE or HMR2.0. The overall design of InitNet is shown in Fig. 6. QuaMo initial pose q_0 , translation r_0 , and SMPL shape β are directly learned from the inputs obtained from the 3D pose estimator. The shape β is kept constant throughout the whole 100-frame sequential integration of QuaMo. The initial angular velocity ω_0 and linear velocity v_0 are linearly mapped from the error between the first two reference poses $\hat{q}_1 \otimes \hat{q}_0^*$ and the first two root translations $\hat{r}_1 - \hat{r}_0$.

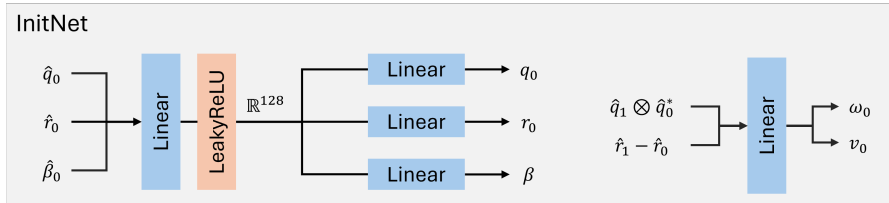


Figure 6: The architecture of InitNet. The initial solution of QuaMo is predicted based on the first two frames of the 3D pose estimator (TRACE or HMR2.0). To enforce the shape plausibility, β is kept constant throughout the sequence.

B ADDITIONAL RESULTS

In this section, we present additional qualitative results of QuaMo, together with the corresponding 2D projection on the input images. For each example presented in Fig. 7, the input image is on the left, 3D motion estimation is on the right, and the projected 2D poses are obtained with the provided intrinsic matrix from their respective datasets. For more insights into QuaMo, we encourage the reader to have a look at our videos that can be viewed by accessing the *index.html* file.

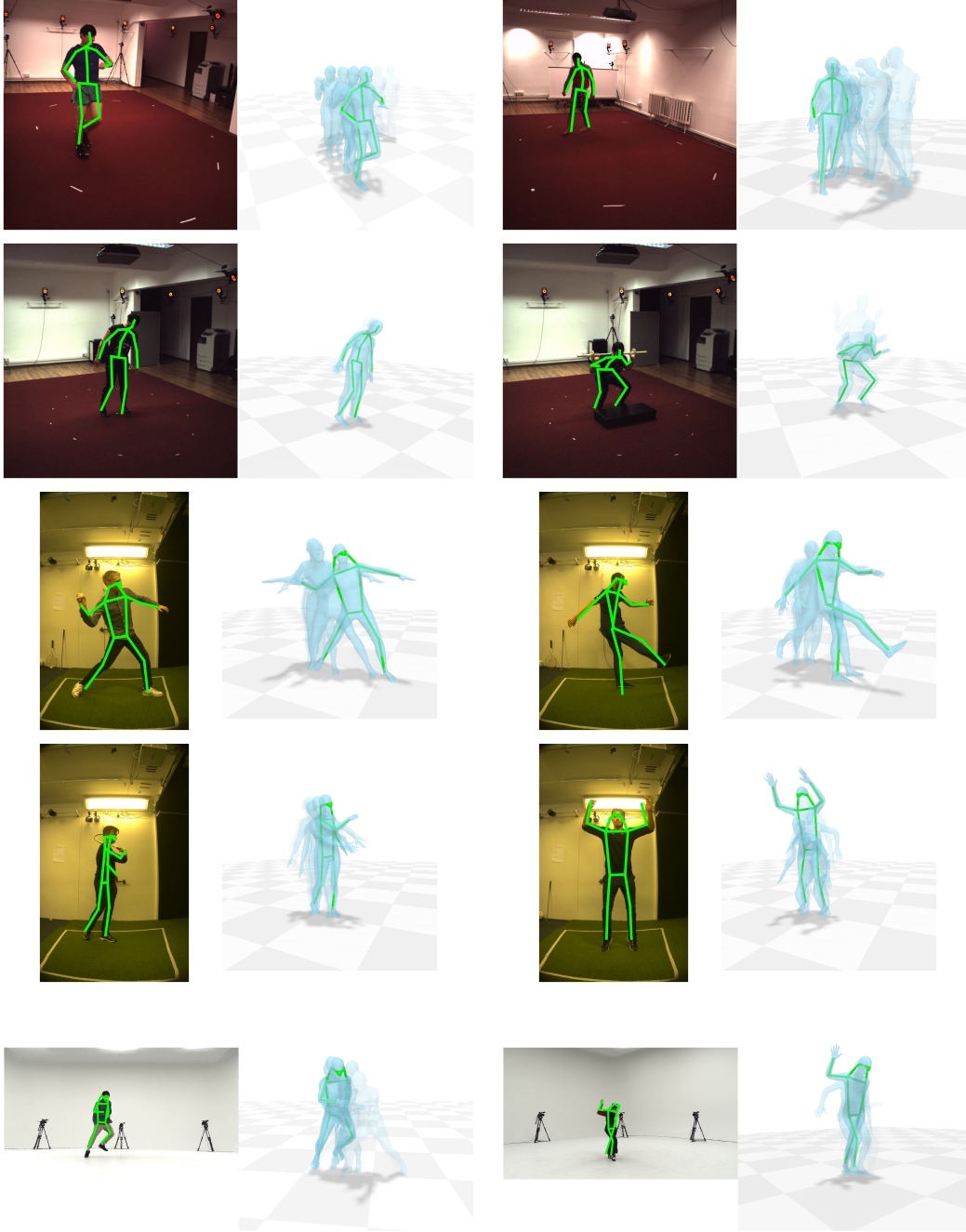


Figure 7: Additional qualitative results on Human3.6M (first row), Fit3D (second row), SportsPose (third and fourth rows), and AIST (last row). The transparency of 3D poses corresponds to their time stamps in the motion sequence. The 2D projections (2D green) is obtained by multiplying camera intrinsic matrix with 3D keypoints (3D green). The Human3.6M 17 keypoints configuration is used for Human3.6M and Fit3D, while the COCO 17 keypoints used for SportsPose and AIST.

C TRAINING DETAILS ON AIST

Unlike the optimization-based approach DiffPhy, our QuaMo is learning-based; therefore, we train QuaMo on 10 random samples from the provided training data of AIST, which are different from the 15 test sequences suggested by Gärtner et al. (2022a). To keep a fair comparison, we also train and test on only the first 120 frames of the respective sequences. The details about the selected training sequences can be found in Tab. 5.

Sequence	Frames
gHO_sFM_c01_d19_mHO3_ch04	1-120
gKR_sFM_c01_d28_mKR3_ch04	1-120
gLH_sFM_c01_d16_mLH3_ch04	1-120
gMH_sBM_c02_d24_mMH3_ch03	1-120
gMH_sBM_c03_d24_mMH3_ch06	1-120
gMH_sBM_c04_d24_mMH3_ch01	1-120
gMH_sBM_c05_d24_mMH3_ch07	1-120
gMH_sBM_c06_d24_mMH3_ch05	1-120
gMH_sBM_c08_d24_mMH3_ch04	1-120
gMH_sBM_c09_d24_mMH3_ch09	1-120
gMH_sFM_c07_d24_mMH3_ch18	1-120

Table 5: Dancing samples from the AIST dataset are used for training.

D COMPUTATIONAL COMPLEXITY

QuaMo is designed to be a real-time approach that takes in single-frame input and outputs next frame estimation. The number of parameters of InitNet is 62361, and ControlNet is 559074. In total, QuaMo has 621435 learnable parameters, which is significantly lower than 7.2M parameters of the competitive approach OSDCap. The per-frame processing time of QuaMo is **5.74ms** on the NVIDIA A100, 7.15ms on the NVIDIA Tesla T4, significantly faster than ≈ 25 ms of OSDCap.

E APPROXIMATION OF QUATERNION INTEGRATION

Prior work, *i.e.* NeurPhys or OSDCap, approximate the next quaternion solution q_{t+1} given the current quaternion q_t and the current angular velocity ω_t via the following equations

$$\begin{aligned}
 \dot{q}_t &= q_t \otimes \left(0, \frac{1}{2} \omega_t \right), \\
 q_{t+\Delta t} &= q_t + \dot{q}_t \Delta t, \\
 q_{t+\Delta t} &= q_{t+\Delta t} / \|q_{t+\Delta t}\|.
 \end{aligned} \tag{7}$$

The integration scheme in Eq. 7 essentially moves the quaternion outside the sphere S^3 and thus requires the magnitude renormalization. This introduces high error during integration, especially for long human motion sequences. The correct method for computing the next quaternion solution is addressed in Sec.3.2 of the main paper, and it effectively reduces the error of captured motions.

F GLOBAL TRANSLATION ANALYSIS

We plot two examples of root translation in world coordinate in Fig. 8. The root translation is computed via Euler’s integration, described in Sec.3.3 with $\Delta t = 0.04$, the real-time transitioning of a motion capture sequence of 25Hz. Previous work, such as NeurPhys Shimada et al. (2021), splits the integration into six iterations, causing the numerical errors from Euler’s method to build up and accumulate over each iteration, while QuaMo does not have this problem. Fig. 8 clearly demonstrates the performance gains that QuaMo provides, in both accuracy and smoothness with respect to the ground truth trajectory.

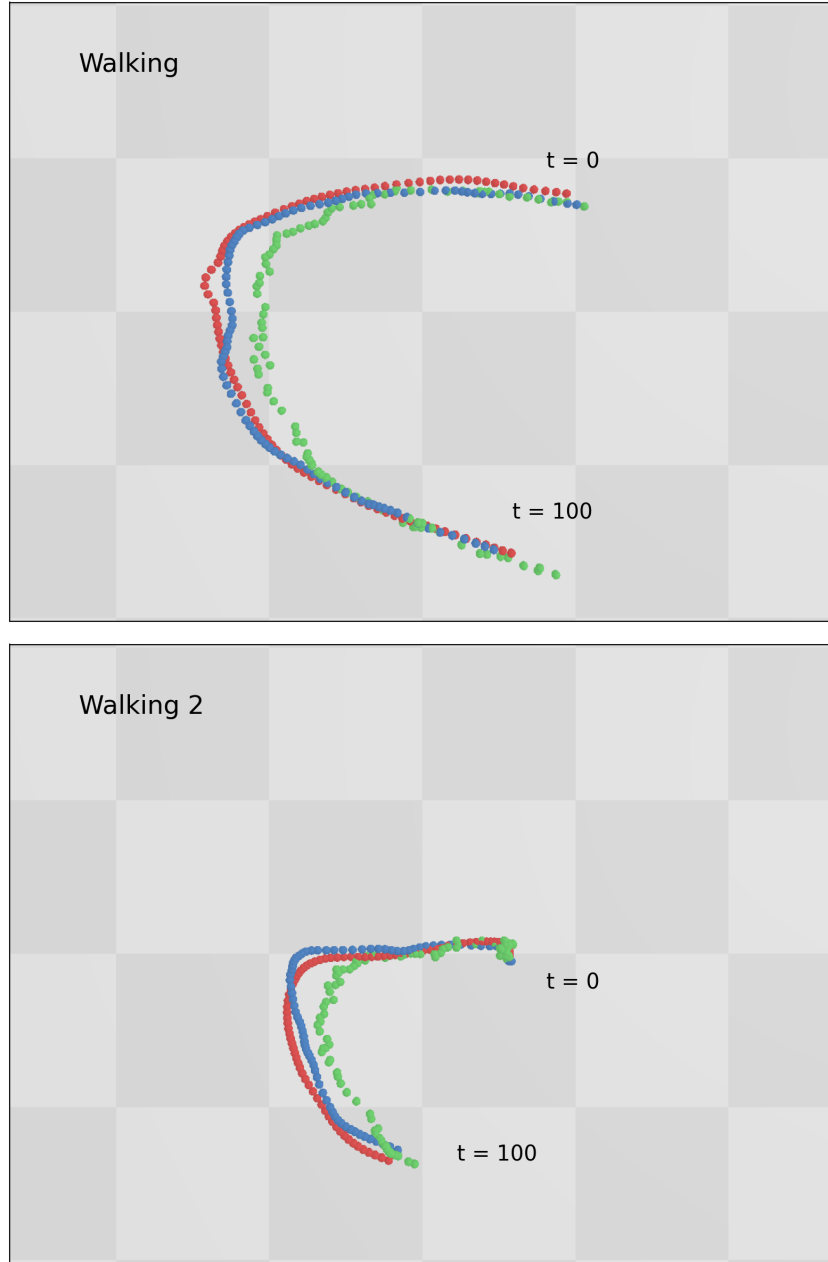


Figure 8: Root trajectory comparison between QuaMo (blue), input signals from TRACE (green), and ground truth (red). QuaMo provides a highly smooth and accurate root trajectory estimation, with no numerical error.