Learning Motion Priors for 4D Human Body Capture in 3D Scenes
Appendix

A. Architecture Details

The model architecture for motion priors is illustrated in Fig. S1. The motion smoothness prior and the motion infilling prior share a similar network architecture, The encoder includes 5 consecutive convolution blocks, with each block containing [conv3x3, LeakyReLU, conv3x3, LeakyReLU, MaxPooling] layers. The motion smoothness prior has the feature channel of [32, 64, 64, 64, 64] for the output of each encoder block. The motion infilling prior has the feature channel of [32, 64, 128, 256, 256] for the output of each encoder block. The decoder includes 5 deconvolution blocks, with each block containing [deconv3x3, LeakyReLU, deconv3x3, LeakyReLU] layers. For the motion smoothness prior, since the smooth constraint (Eq. 3) works on the latent space, we do not downsample the features so that the latent space can preserve the full spatial-temporal resolution the same as the input motion, to model smooth full-body dynamics without losing motion details, thus the MaxPooling layer is not included in the motion smoothness prior.

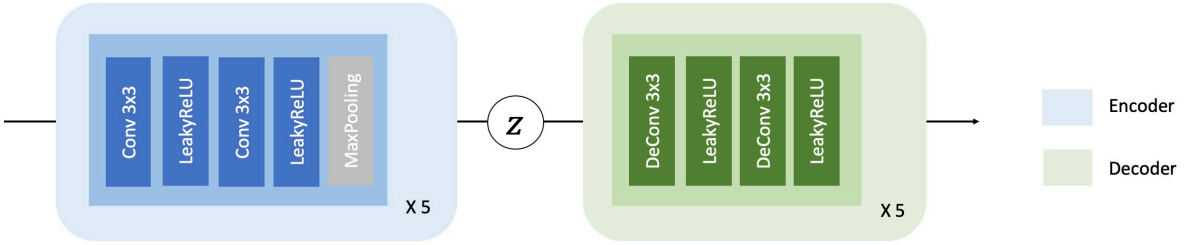

Figure S1: Model architecture for motion priors. The encoder includes 5 consecutive [conv3x3, LeakyReLU, conv3x3, LeakyReLU, MaxPooling] blocks, and the decoder includes 5 consecutive [deconv3x3, LeakyReLU, deconv3x3, LeakyReLU] blocks. The MaxPooling layers are only included in the motion infilling prior network.

B. Experiment Details

B.1. Implementation Details

The proposed algorithm is implemented with PyTorch 1.4.0. We use a single TITAN RTX GPU for training and optimization. For the motion smoothness prior and motion infilling prior training, we use ADAM as the optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with the learning rate 1e-4. The motion smoothness prior is trained for 150 epochs with a batch size of 60, and the motion infilling prior is trained for 900 epochs with a batch size of 120. For the proposed multi-stage optimization pipeline on PROX [19], we use ADAM as the optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with the learning rate 5e-3. In both Stage 2 and Stage 3, we optimize for 900 steps for each motion clip of 100 frames.

B.2. Marker Placement

The body surface marker placement is illustrated in Fig. S2. We choose 67 markers on the SMPL-X body mesh surface, following the SSM2 marker setting in [38]. Furthermore, we select additional 14 markers on the face and fingers to enforce smoothness over hand motions and facial expressions. Note that the additional 14 markers are only utilized in the motion smoothness prior, as we mainly focus on motion infilling for lower part of the body in the motion infilling prior. Compared with body joints, body surface markers can better model degrees-of-freedom (DoFs) and incorporate body shape information [71].

B.3. Evaluation Details

Power spectrum KL divergence (PSKL). We use PSKL to measure the distribution distance between two datasets, as in [20]. Formally, given a motion sequence of T frames, and each frame represented by F features, the power spectrum of each feature sequence s_f is $PS(s_f) = ||FFT(s_f)||^2$. x, y, z accelerations of each frame are used as the features and $F = 3M$, where M denotes the number of body markers or joints. The average power spectrum for feature f over N motion

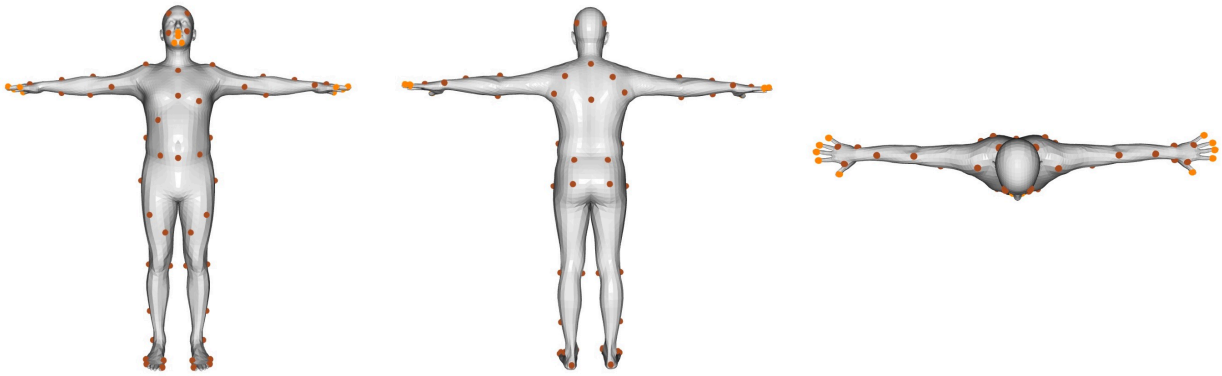

Figure S2: Marker placement for motion priors. The body markers are denoted by spheres over the SMPL-X body surface, following the marker setting (brown) in [38], with 14 additional markers (orange) on the face and fingers. From left to right: the front view, the back view and the top view.

sequences on a dataset C is computed as:

$$PS(C|f) = \frac{1}{N}\sum_{n=1}^{N} PS(s_f), \tag{10}$$

and $PS(C|f)$ is normalized along the frequency dimension. PSKL between datasets C and D is the average power spectrum KL divergence over all feature dimensions:

$$PSKL(C,D) = \frac{1}{F}\sum_{f=1}^{F}\sum_{e=1}^{E} ||PS(C|f)|| * log(\frac{||PS(C|f)||}{||PS(D|f)||}), \tag{11}$$

where e is frequency. KL divergence is asymmetric, thus both directions PSKL(C,D) and PSKL(D,C) are computed.

Motion infilling prior experiments. Here we describe the fitting procedure for the motion infilling prior evaluation on AMASS [38] in detail. For both our proposed method and the baseline method, firstly we implement per-frame fitting with the following objective function:

$$E_{PF} = E_{3D} + E_{prior}, \tag{12}$$

where E_{3D} is the L1 loss between infilled marker positions inferred by the infilling network and marker positions of the SMPL-X body to optimize, and E_{prior} is the prior term for body pose and hand pose. The per-frame fitting aims to provide a good initialization for the temporal fitting. Initialized from per-frame fitting results, the temporal fitting is implemented by minimizing:

$$E_{TF} = E_{3D} + E_{prior} + E_{smooth} + E_{foot}, \tag{13}$$

where E_{smooth} is the proposed smooth prior term in Eq. 3. For our proposed method, E_{foot} is the second term in Eq. 8, which penalizes foot vertex velocity according to the predicted foot-ground contact states. For the baseline method, E_{foot} is defined by a heuristic cue:

$$E_{foot} = \sum_{k,t:z_k^t \leq z_{thres}} d(v_k^t, a), \tag{14}$$

where z_k^t is the distance from the ground of foot joint k at frame t, with z_{thres} set to 10cm. v_k^t is the absolute velocity magnitude of foot joint k at frame t. $d(v_k^t, a)$ corresponds to $|v_k^t - a|$ if $v_k^t \geq a$, 0 otherwise. Foot velocity threshold a is set to 10cm/s. This term penalizes foot joint velocity when its distance to the ground is smaller than 10cm.

Table S1: **Ablation study for swapping the proposed Stage 2 and Stage 3 on PROX.** PSKL-M and PSKL-J denote PSKL computed on markers and joints, respectively. (P, A) denotes PSKL(PROX, AMASS), and (A, P) the reverse direction. For each metric, the better result is in boldface.

	2DJE ↓	PSKL-M ↓		PSKL-J ↓		NonColl ↑
Methods		(P,A)	(A,P)	(P,A)	(A,P)	
Ours-S3	**20.23**	**0.236**	**0.234**	**0.256**	0.255	**0.979**
Ours-S3-S2	20.64	0.273	0.236	0.307	**0.254**	0.972

Table S2: **Comparison with regression-based denoising models on PROX.** 2DJE denotes the 2D joint accuracy. PSKL-J denotes PSKL of joints. (P, A) denotes PSKL(PROX, AMASS), and (A, P) the reverse direction. For each metric, the best result is in boldface.

Methods	2DJE ↓	PSKL-J (P,A) ↓	PSKL-J (A,P) ↓
Wang et al. [62]	140.47	0.294	0.303
Holden et al. [21]	62.97	0.487	0.462
Kim et al. [30]	66.05	0.285	0.278
Ours-SP	**20.64**	**0.272**	**0.275**

Swap Stage 2 & Stage 3. As the motion infilling prior is trained with high-quality data on AMASS, and the proposed self-supervised test fine-tuning relies on the motion of visible body parts, it requires smooth motions as input for good performance. Therefore we first recover temporal consistent motion by the Stage 2, and then include the motion infilling prior in Stage 3. As shown in Tab. S1, if we swap Stage 2 and Stage 3 (denoted by 'Ours-S3-S2'), the overall motion naturalness (PSKL score) will degrade, as well as the pose accuracy (2DJE).

C. Comparison with Regression-based Methods

On PROX, we additionally compare the proposed motion smoothness prior (Ours-SP) with three regression-based denoising methods [21, 30, 62]. These methods directly output smooth motions represented by body joints by taking noisy motion as input. For a fair comparison, we train them on AMASS, adding Gaussian noise to the input motion.

Tab. S2 shows that our smoothness prior (Ours-SP) achieves significantly higher joint accuracy, and produces more realistic motions according to the PSKL scores. The regression-based methods are trained with synthesized noise, which limits their generalizability to different noise distributions, and frequently produces inaccurate global reconstruction, while our motion prior is trained with clean motions, and works very well both on 3DPW captured by IMU sensors and PROX captured by Kinect. Besides, it is unclear how to incorporate the 3D scene constraints directly into the regressors. In contrast, our motion priors and human-scene interaction constraints can be unified in an optimization framework to produce realistic motions that satisfy 3D scene constraints.